VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Zabezpečenie IP telefónnej infraštruktúry s využitím pokročilých systémov IPS

Securing IP Telephony Infrastructure Using Advanced IPS Systems

Bc. Michaela Píšová

Diplomová práca, Školiteľ: Ing. Filip Řezáč, Ph.D., Ostrava 2020/2021

Abstrakt

Cieľom diplomovej práce bolo otestovať nástroj Suricata v režime prevencie narušenia systémov pri realizácii útokov na pobočkovú ústredňu Asterisk, pre ktorú boli nakonfigurované dva softvérové telefóny Jitsi a PhonerLite. Pri testovaní nástroj Suricata najskôr pracoval v spojení s filtračným nástrojom IPtables a neskôr s jeho novším nástupcom NFtables. Pri realizácii testovania boli použité útoky typu SIPVicious, Inviteflood a skenovanie siete pomocou príkazu NMAP. V práci som sa venovala tvorbou pravidiel pre zachytávanie anomálií s cieľom získať výsledky z pohľadu úspešnosti zablokovania potenciálnych útokov so zameraním na IP telefóniu.

Kľúčové slová: Voice over IP, systém prevencie narušenia IPS, Suricata, IPtables, Nftables, NFqueues, PBX Asterisk

Abstract

The aim of the diploma thesis was to test the Suricata tool in the mode of prevention of systems intrusion in the implementation of attacks on the Asterisk branch exchange, for which two software phones Jitsi and PhonerLite were configured. In testing the Suricata tool first worked in conjunction with the IPtables filter system and later with its newer successor, NFtables. SIPVicious, Inviteflood, and network scanning with the use of NMAP command were used to perform the testing. In my work I focused on creating rules for capturing anomalies in order to obtain results in terms of the success of blocking potential attacks with a focus on IP telephony.

Key Words: Voice over IP, Intrusion Prevention System IPS, Suricata, IPtables, Nftables, NFqueues, PBX Asterisk

Poďakovanie

Chcela by som sa poďakovať vedúcemu práce Ing. Filipovi Řezáčovi, Ph.D. za jeho cenné rady, odbornú pomoc a ochotu pri spracovaní práce.

Obsah

Zoznam obrázkov

Zoznam skratiek

Access Point (AP)
Acknowledge (ACK)
Address Resolution Protocol (ARP)
Application Programming Interface (API)
Central Processing Unit (CPU)
ContributingSsource IDs (CSRC)
Coronavirus disease 2019 (COVID)
Denial-of-Service (DoS)
Domain Name System (DNS)
Dynamic Host Configuration Protocol (DHCP)
File Transfer Protocol (FTP)
Host Intrusion Protection System (HIPS)
Hypertext Transfer Protocol (HTTP)
Identification Number (ID)
Internet Control Message Protocol (ICMP)
Internet Engineering Task Force (IETF)
Internet Protocol (IP)
Intrusion Detection System (IDS)
Intrusion Protection System (IPS)
Network Address Translation (NAT)
Network Behavior Analysis (NBA)
Network Intrusion Protection System (NIPS)
Man-in-the-Middle (MITM)
Media Access Control (MAC)
Open Security Foundation (OISF)
Open Systems Interconnection Reference Model (OSI)
Payload Type (PT)
Private Branch Exchange (PBX)
Public Switched Telephone Network (PSTN)
Quality of Service (QoS)
Real Time Transport (RTP)
RTP Control Protocol (RTCP)
Secure Shell (SSH)
Security Event Manager (SEM)
Session Initiation Protocol (SIP)
Spam Through Internet Telephony (SPIT)
Stream Control Transmission Protocol (SCTP)
Synchronization Source Identifier (SSRC)
Synchronize (SYN)
Time to Live (TTL)
Transmission Control Protocol (TCP)
User Agent (UA)

User Datagram Protocol (UDP)
Voice over IP (VoIP)
Voice over IP Security Alliance (VoIPSA)
Wireless Intrusion Prevention System (WIPS)

Úvod

Vývoj hlasových služieb Voice over IP (VoIP) prináša čoraz väčšiu pozornosť v telekomunikačnom priemysle, pretože umožňuje firmám a jednotlivým užívateľom používať jednoduchšiu hlasovú IP telefóniu vďaka jej pohodlnosti a nákladovej efektívnosti. Aplikácia VoIP, ako je IP telefónny systém zahŕňa odosielanie hlasových prenosov ako dátových paketov cez súkromné alebo verejné IP siete. Architektúra tejto služby posúva inteligenciu smerom ku koncovým zariadeniam a tak poskytuje príležitosť na vytvorenie mnohých nových služieb, ktoré pri tradičnom telefónnom systéme nemôžeme predpokladať. Technológia VoIP je oveľa viac atraktívna voľba pre prenos hlasu v porovnaní s tradičnou technológiou prepínania okruhu napríklad z dôvodu, že jej stačia nízke náklady na vybavenie, integráciu hlasových a dátových aplikácií, nižšie požiadavky na šírku pásma a podobne [1].

Vzhľadom na to, že sa zvyšuje využitie VoIP, zvyšuje sa aj potenciálne riziko hrozby pre bežného užívateľa. V dnešnom svete existuje rozsiahle množstvo kategórií hrozieb a útokov, ktoré skúmajú spoločnosti Voice over IP Security Alliance (VoIPSA) a Internet Engineering Task Force (IETF) [2].

Aby sme zabránili pokusom o zneužitie zraniteľných miest v systémoch alebo aplikáciách, môžeme použiť systém prevencie narušenia systému (IPS), pomocou ktorého monitorujeme zariadenie, o ktorého bezpečnosť máme záujem a tento systém teda vykonáva potrebné kroky na zablokovanie všetkého, čo vyhodnotí, že pre náš chránený systém predstavuje hrozbu.

V systémoch prevencie narušenia môžeme pre monitorovanie a sledovanie sieťového prenosu použiť niektoré nástroje, ktoré sú založené na operačnom systéme Linux ako sú napríklad IPtables, NFqueue a Suricata, pomocou ktorých môžeme nastavovať pravidlá filtrovania IP paketov, formovať alebo manipulovať prenos.

Prvá kapitola diplomovej práce je zameraná na predstavenie hlasovej siete VoIP. V skratke popisuje dva základné protokoly, ktoré sa používajú v spojení s VoIP technológiami, a teda protokol RTP pre komunikáciu v reálnom čase a protokol SIP, ktorý je používaný ako signalizačný protokol určený na vytváranie a ukončovanie multimediálnych spojení.

Druhá kapitola popisuje a klasifikuje hrozby, s ktorými sa stretávame v službách VoIP podľa štandardu VOIPSA. Pre jednotlivé klasifikácie sú uvedené najčastejšie typy hrozieb, ktoré sa v sieti bežne vyskytujú.

Tretia kapitola diplomovej práce sa zameriava na systém prevencie narušenia IPS, ktorého hlavnou úlohou je identifikácia potenciálnych hrozieb, ktoré sa nachádzajú v systémoch alebo aplikáciách. Táto kapitola tiež popisuje možné metódy detekcie systému prevencie narušenia, jeho základnú architektúru a jeho základné typy. Okrajovo popisuje tri najčastejšie používané protokoly pre systémy detekcie a prevencie narušenia.

Štvrtá kapitola sa zameriava na nástroje, ktoré umožňujú detekciu sieťových hrozieb pomocou nástroja Suricata, filtrujúce sieťový prenos pomocou systému Iptables a Nftables, využívajúce metódu NFqueue. V tejto kapitole sú definované jednotlivé tabuľky, ciele, moduly, príkazy a pravidlá pre filtračný systém IPtables, a porovnáva IPtables s jeho nástupcom NFtables. Táto kapitola tiež obsahuje popis základnej architektúry a štruktúry nástroja Suricata, jeho princíp funkcie v režime IPS a tiež jeho prácu s nástrojom NFqueue.

Piata kapitola diplomovej práce predstavuje testovaciu topológiu siete, pre ktorú budú realizované útoky pomocou vhodných nástrojov na IP telefóny. Detailne popisuje inštaláciu

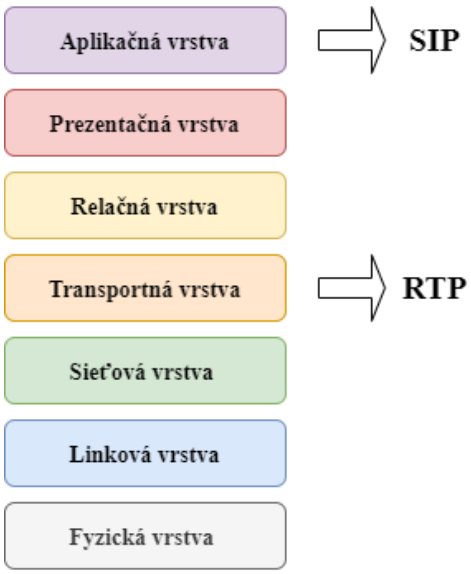
pobočkovej ústredne Asterisk a nastavenie softvérových telefónov, komunikujúcich prostredníctvom Asterisk PBX.

Šiesta kapitola je zameraná na realizáciu praktických útokov typu SIPVicious, Inviteflood a na skenovanie siete pomocou NMAP na pobočkovú ústredňu Asterisk a jeho IP telefónov. Pri testovaní útokov nástroj Suricata pracuje v režime prevencie narušenia systémov, pričom v prvej časti pokusov pracuje v spojení s filtračným systémom IPtables, a v druhej časti pokusov pracuje v spojení s NFtables. V oboch prípadoch sa snažia jednotlivé pokusy o podozrivú komunikáciu zablokovať. Jednotlivé výsledky realizácie praktickej časti sú zaznamenané v tejto kapitole.

1. Technológia Voice over IP

Voice over IP (VoIP) je technológia, ktorá umožňuje realizovať hlasové hovory cez siete založené na Internetovej technológii, a to konkrétnejšie na protokole IP [29]. Služby, ktoré poskytuje, sú podobné ako u bežných pevných telefónov, ale všeobecne ponúka lacnejšie riešenie. Požívaná je hlavne na digitálnu komunikáciu a umožňuje nám naviazať jednoduché telefónne spojenie pomocou Internetového pripojenia, pričom prevádza hlasové (analógové) signály do digitálneho signálu (pakety dát), ktoré sa pohybujú cez verejnú a súkromnú sieť Internetového protokolu [31]. VoIP využíva pre prenos hlasu technológiu prepínania paketov, ktorá na uľahčenie hovorov používa niekoľko protokolov.

Medzi dva základné protokoly patrí protokol Real Time Transport (RTP), ktorý definuje štandardný formát paketu na doručovanie médií cez Internet a nachádza sa na štvrtej transportnej vrstve The Open Systems Interconnection (OSI) modelu. Druhým protokolom je Session Initiation Protocol (SIP), ktorý je používaný ako signalizačný protokol pre vytvorenie, udržanie a uskutočňovanie relácie medzi dvomi alebo viacerými účastníkmi [30]. Nachádza sa na poslednej, siedmej vrstve aplikačnej vrstvy referenčného OSI modelu.



Obrázok 1 Model OSI pre protokol RTP a SIP

1.1 Real Time Transport (RTP)

Prenosový protokol v reálnom čase RTP je protokol, ktorý zaisťuje multimediálny prenos v reálnom čase. Takmer všetky implementácie VoIP používajú protokol RTP na prepravu médií v spojení so signalizačným protokolom SIP. V skutočnosti nezaručuje správne doručenie dát ani správne poradie jednotlivých paketov, ale definujú ich poradové čísla, podľa ktorých môžu multimediálne aplikácie rozpoznať chýbajúce pakety.

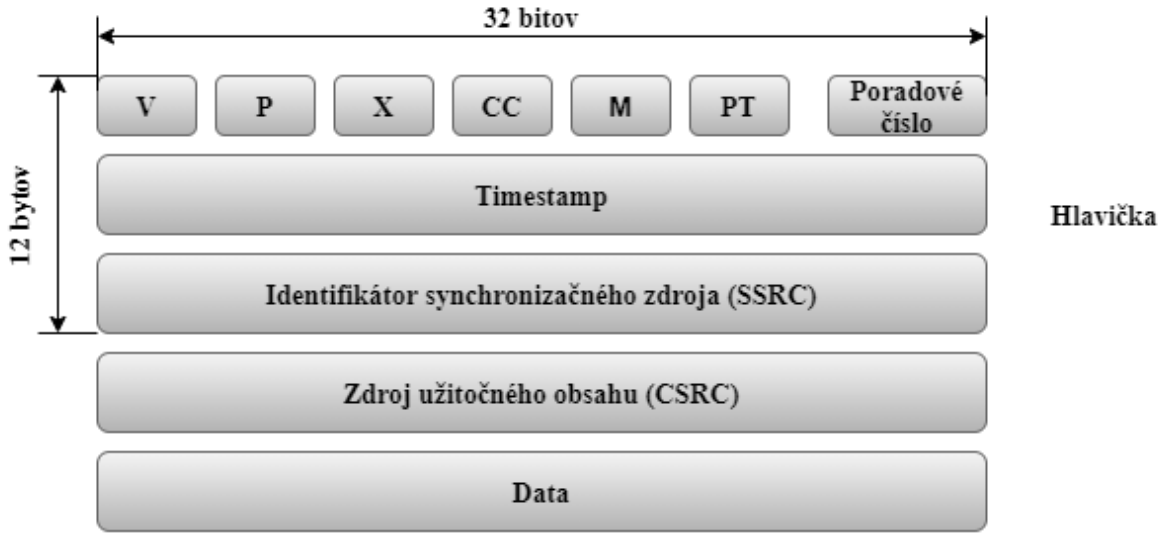
Najčastejšie používa protokol User Datagram (UDP), pretože umožňuje rýchlejšie dodanie údajov, ale môže tiež používať aj iné protokoly [32]. Prenosový protokol v reálnom čase poskytuje:

- identifikáciu typu užitočného zaťaženia,
- identifikáciu zdroja,
- poradové číslovanie,
- časové pečiatky.

Prenos dát RTP je rozšírený o kontrolný protokol RTCP, ktorý poskytuje účastníkom relácie RTP spätnú väzbu o kvalite distribúcie dát. Jeho hlavnou funkciou je napríklad monitorovanie a kontrola preťaženia QoS, zhromažďovanie informácií ako napríklad počet odoslaných bajtov, paketov, počet stratených paketov, jitter, spätnú väzbu a dobu odozvy [33].

1.1.1 Datagram RTP

Protokol RTP bol navrhnutý pre individuálne aj skupinové prenosy, ako aj pre jednosmerný a obojsmerný prenos. K multimediálnemu obsahu RTP pripája záhlavie, ktoré obsahuje jednotlivé informácie znázornené na obrázku 2.



Obrázok 2 Formát dátového RTP datagramu

***Verzia* (V)** RTP protokolu obsahuje dva bity.
Výplň (P), jeden bit, v prípade, že je tento bit nastavený, tak je na konci paketu jeden alebo viac bytov, ktoré nie sú súčasťou užitočného obsahu.
Rozšírenie (X), jeden bit, ak je tento bit nastavený, tak nasleduje za pevným záhlavím jedno rozšírené záhlavie. Umožňuje vložiť rozširujúce informácie do záhlavia RTP paketu.
Počet CSRC (CC), štyri bity, definuje počet CSRC identifikátorov, ktoré nasledujú za pevným záhlavím. V prípade, že je počet SCRC nula, zdrojom synchronizácie je zdroj užitočného obsahu.
Záložka (M), jeden bit, je záložkový bit, definovaný konkrétnym profilom média.

Typ užitočného zaťaženia (PT), sedem bitov, je index z tabuľky profilu média, ktorý popisuje formát užitočného obsahu.
Poradové číslo (Sequence number), šestnásť bitov, definuje jedinečné číslo paketu, ktoré popisuje pozíciu paketu v poradí paketov.
Časová známka (Timestamp), tridsaťdva bitov, vyjadruje moment odobratia vzorky prvého bytu užitočného obsahu.
Synchronizačný zdroj (SSRC), tridsaťdva bitov, identifikuje synchronizačný zdroj.
Zdroj užitočného obsahu (CSRC), tridsaťdva bitov, identifikuje zdroj prospievajúci do užitočného obsahu.

1.2 Signalizačný protokol SIP

Protokol SIP je riadiaci protokol na aplikačnej vrstve, určený na vytváranie, upravovanie a ukončovanie multimediálnych relácií, ako napríklad telefonické hovory cez Internet, alebo tiež umožňuje pozývať účastníkov na už existujúce relácie, ako napríklad na multicastovú konferenciu. Môže byť prenášaný cez protokol riadenia prenosu TCP, užívateľský datagramový protokol UDP alebo protokol prenosu riadenia toku SCTP. Klienti SIP zvyčajne využívajú pre protokol TCP/UPD port číslo 5060 alebo port číslo 5061 pre prenos SIP na servery a iné koncové body. Protokol SIP vôbec nebol navrhnutý tak, aby bol bezpečný, preto je veľmi ľahko napadnuteľný, ale napriek tomu patrí k jednému z najdôležitejších protokolov hlasovej signalizácie v hlasových sieťach VoIP.

Medzi jeho päť najdôležitejších schopností naviazania a ukončenia multimediálnej komunikácie patrí umiestnenie užívateľov v relácií, dostupnosť užívateľa, možnosti užívateľa, nastavenie relácie a riadenie relácie. Architektúra protokolu SIP je založená na type klient-server, čo znamená, že komunikácia prebieha výmenou správ *žiadosť* a *odpoveď na žiadosť* [35].

1.2.1 SIP správy

SIP je textovo orientovaný protokol, pre ktorý existujú dva rôzne typy SIP správ, a to požiadavka od klienta na server alebo odpoveď zo servera na klienta. SIP správy sú typu *žiadosť* a *odpoveď.*

Žiadosti sú odosielané od SIP klienta na SIP server. Medzi základné žiadosti patria:

INVITE metóda, ktorá je používaná pre iniciovanie a aktualizáciu multimediálnej relácie.
ACK metóda, ktorá je používaná v spojení s metódou *INVITE* a potvrdzuje, že klient prijal odpoveď.
CANCEL metóda umožňuje ukončiť transakciu *INVITE* ešte predtým, ako iniciátor volania hovor prijme.
BYE metóda umožňuje ukončiť klientovi reláciu iniciovanú metódou *INVITE*.
REGISTER metóda informuje server o tom, že sa v sieti nachádza klient, ktorý je dostupný na komunikáciu.
OPTIONS metóda je používaná na to, aby sme boli schopní obdržať informácie o potenciálnej alebo existujúcej relácii.

Odpovede posiela SIP server klientovi ako odpoveď na žiadosť; odpovede sú delené do šiestich tried, a to:

1xx: Predbežná, ktorá príjme žiadosť a pokračuje v spracovaní.
2xx: Úspešná, teda žiadosť bola úspešne prijatá, pochopená a akceptovaná.
3xx: Presmerovanie, vyžaduje vykonať ešte ďalšie kroky, aby bolo možné dokončiť žiadosti.
4xx: Chyba klienta, žiadosť obsahuje zlú syntax, alebo žiadosť nemôže byť na tomto serveri.
5xx: Chyba servera, server pravdepodobne nevykonal platnú žiadosť.
6xx: Celková chyba, žiadosť nie je splnená pre žiadny server [34].

2. Útoky a hrozby

Útoky v sieťach VoIP môžu znamenať, že akákoľvek služba môže byť narušená, odmietnutá alebo zmenená tak, že pôvodná služba už viac nebude považovaná za dôveryhodnú alebo dostupnú. Pre jednotlivé útoky existujú potenciálne hrozby, ktoré v sieťach VoIP boli rozdelené podľa VOIPSA na obrovskú klasifikáciu hrozieb. Môžeme tvrdiť, že akýkoľvek prvok vrátane podporných komponentov alebo protokolov v nasadení VoIP môže spôsobiť zraniteľné miesta a je teda ťažké chrániť každé VoIP nasadenie. Hrozby, ktoré sú uvedené v IETF pre VoIP bezpečnostné hrozby, sú také, ktoré by sme mali brať do úvahy pri návrhu protokolu. Prvá verzia konceptu IETF obsahovala zoznam nasledujúcich kategórií hrozieb:

- hrozby zachytenia a modifikácie,
- hrozby prerušenia služby,
- hrozby zneužitia služby,
- sociálne hrozby.

V súčasnej dobe existuje veľké množstvo rôznych kategorizácií a rôzne klasifikácie majú rôzne účely. VOIPSA veľmi podrobne analyzuje hrozby, aby poskytovala čo najviac informácií a tiež pomáha užívateľom pochopiť súvisiace hrozby. Klasifikácia hrozieb IETF kategorizuje hrozby na základe toho, ako je možné vylepšiť špecifikácie protokolu tak, aby sa minimalizoval dopad útoku a preto neberie do úvahy problémy spojené s podpornou infraštruktúrou, ako sú platformy operačného systému a konfigurácie siete [3]. Hrozby spojené s VoIP je možné rozdeliť do nižšie uvedených kategórií:

- prerušenie služby a obťažovanie,
- odpočúvanie,
- maskovanie a odcudzenie identity,
- nepovolený prístup,
- podvod.

2.1 Prerušenie služby a obťažovanie

Útoky v kategórii ***prerušenie služby*** dokážu ovplyvniť akýkoľvek sieťový prvok, ktorý podporuje službu VoIP, vrátane smerovačov, DNS serverov, SIP proxy a podobne. Tieto útoky môžu byť iniciované na diaľku, bez priameho prístupu k prvkom cieľovej siete alebo manipulácie s protokolmi VoIP. Útočník sa zameriava na okrajové zariadenie ako napríklad na telefón VoIP, alebo na základnú sieť komponentov alebo súborov komponentov, ako napríklad SIP proxy, ktoré môžu mať vplyv na komunitu užívateľov.

Medzi najčastejší typ útoku prerušenia služby patrí *Denial-of-Service (DoS)*, ktorý dokáže ovplyvniť akúkoľvek sieťovú službu založenú na IP. Dopad útoku DoS sa môže pohybovať od miernej degradácie služby, až po jej úplnú stratu. Je to typ útoku na Internetové služby alebo stránky, kde cieľom je danú službu zneužiť a zneprístupniť ju ostatným užívateľom [4].

Útok typu *Denial-of-Service (DoS)* môžeme klasifikovať nasledovne:

- *DoS s chybnou požiadavkou*, kde útočník vytvoril požiadavku alebo odpoveď na SIP, ktorá zneužíva zraniteľnosť v SIP proxy serveri, kde používajú správy typu INVITE na zahltenie proxy serveru, alebo použijú SIP UA, čo vedie k úplnej alebo čiastočnej strate funkcie. Útočníci tiež dokázali, že IP implementácie niektorých pevných telefónov sú zraniteľné voči útokom fregmentácie a útokom DoS na báze DHCP, čo má za následok to, že pre zariadenia VoIP je dôležitá ochrana infraštruktúry.

- *DoS na základe zaťaženia*, kde v tomto prípade útočník nasmeruje veľké objemy prenosu na cieľ alebo skupiny cieľov a pokúsi sa vyčerpať zdroje, ako napríklad čas spracovania CPU, šírku pásma siete alebo pamäte [5].

Medzi základne typy DoS útokov patrí:

SYN flood – útočník posiela postupne veľa paketov so žiadosťou SYN na cieľový počítač, pričom tento spôsob útoku je zvyčajne neúspešný. Funguje len vtedy, ak server alokuje prostriedky pre nové spojenie ihneď po obdržaní paketu SYN, ešte predtým, ako dostal paket ACK.

Internet Control Message Protocol (ICMP) Flooding – je používaný na IP diagnostiku, chyby a operácie v protokole bez pripojenia.

Teardrop Attacks – zahŕňajú útočníkov, ktorí odosielajú neúplné, dezorganizované a poškodené fragmenty IP a prekrývajú užitočné zaťaženie do cieľového stroja. To má za následok zlyhanie serverov z dôvodu chyby v opätovnej fragmentácii komunikačného kanála.

Peer-to-Peer Attacks – sú druhy distribuovanej siete, v ktorej uzly nazývané „peers“fungujú ako vysielač a prijímač.

Low-Rate Denial-of-Service Attack – ide o útok navrhnutý tak, aby implementoval stratégiu založenú na časovom limite opakovaného vysielania s cieľom znížiť celkový počet TCP manipuláciou s pomalým časovým rozsahom protokolu TCP [6].

Útok typu ***obťažovanie*** predstavuje súhrn rôznych spôsobov, pomocou ktorých útočník obťažuje obeť telefonovaním, ktorému nie je možné zabrániť. Do tejto kategórie môžeme zaradiť aj útok typu SPIT. Ten je nazývaný tiež ako hlasový SPAM, ktorý označuje hromadné, automaticky generované, nevyžiadané hovory. Je podobný e-mailovému spamu. Jednoduchým útokom je vytvorenie skriptu, ktorý iniciuje hovory na široký číselný rozsah alebo rozsah IP adries a odosielala zaznamenanú reč.

Potenciálnym riešením pre tento typ útoku môže byť Blacklisting, obmedzujúci známy spam a Whitelisting, povoľujúci len správnych užívateľov a domény. Ďalším riešením môže byť úprava záznamníka tak, aby zariadenie skutočne odpovedalo na hovor a na základe odpovede na jednoduchú otázku rozhodne, či hovor presmeruje.

2.2 Odpočúvanie

Odpočúvanie je útokom, ktorý je možné vykonať niekoľkými spôsobmi. Útočník môže využívať chyby zabezpečenia, ktoré existujú v protokoloch alebo v softvérovej implementácií komponentov VoIP na zachytenie komunikácie medzi stranami. Komunikáciu medzi užívateľmi, ktorí sa nachádzajú v sieťach PSTN alebo IP je možno útočníkom zachytiť a monitorovať medzi dvoma nič netušiacimi stranami. Tento útok je jednoduchší v sieťach založených na IP kvôli ľahkému prístupu. Najčastejšie sa používa metóda sniffing paketov. Okrem toho za odpočúvanie možno považovať tieto typy útokov:

- analýza prenosu,
- signalizačné odpočúvanie,
- odpočúvanie médií.

Analýza prenosu na linkovej, sieťovej alebo prenosovej vrstve je možná len v prípade, že neexistuje šifrovanie. Pomocou tohto typu útoku môžeme odhaliť informácie o vzoroch hovoru, správaní a zvykoch užívateľa, čo útočníkovi pomáha pri formovaní cieľa. Pri pohľade na paket môžeme vidieť všetky technické údaje súvisiace s komunikáciou a všetky dôverné údaje sú dostupné komukoľvek na dátovej ceste.

V typoch útoku *signalizačné odpočúvanie* a *odpočúvanie médií* získa útočník s prístupom k IP sieti schopnosť monitorovať a zachytávať signalizáciu a mediálne správy medzi dvoma nič netušiacimi stranami.

Paket *sniffing* zachytáva dátové pakety cez počítačovú sieť, pričom útočníci nelegálne získavajú informácie. Tento typ útoku môže byť konfigurovaný dvoma spôsobmi, a to ako *filtrovaný* používaný na to, aby zachytil konkrétne dátové pakety a *nefiltrovaný* používaný v prípade, keď chce útočník zachytiť všetky pakety. Najčastejšie je tento typ útoku realizovaný pomocou techniky útoku Man-in-the-middle (MIMT), prostredníctvom ktorej útočníci využívajú sniffovacie nástroje k zachytávaniu paketov, s ktorými následne manipulujú podľa potreby. Typickým príkladom sniffovacieho nástroja je napríklad WireShark. Tento typ útoku je spomenutý v kapitole 2.4.

2.3 Maskovanie

Maskovanie je typ útoku, pri ktorom má útočník schopnosť vydávať sa za užívateľa, zariadenie, alebo za servis, a to za účelom získania prístupu k sieti, službe, sieťovému prvku alebo informáciám. Špeciálnym prípadom hrozby maskovania je odcudzenie identity, pri ktorej útočník predstiera niekoho identitu, alebo ju niekomu prevezme. Odcudzenie identity sa dá dosiahnuť vzdialenou manipuláciou so signalizačnými alebo mediálnymi tokmi, alebo prostredníctvom neoprávneného prístupu ku komponentom VoIP. Maskovanie útokov v sieťach VoIP môže byť realizované manipuláciou základných protokolov, ktoré poskytujú podporu pre VoIP, ako sú napríklad ARP, IP a DNS.

Najčastejšie je v tomto prípade využívané *falšovanie ID volajúceho*, ktoré je možné vykonávať manipuláciou so správami použitého signalizačného protokolu (napríklad SIP INVITE). Falšovanie ID volajúceho je možné využiť aj v sieťach poskytovateľov VoIP s firemnými zákazníkmi. Poskytovatelia

služieb VoIP vytvárajú konfiguráciu typu Point-to-Point medzi bránou VoIP poskytovateľa a bránou VoIP zákazníka, čím poskytovateľ akceptuje akýkoľvek prenos VoIP bez možnosti overenia ID informácií volajúceho z pôvodnej podnikovej siete. Útočník, ktorý sa nachádza v podnikovej sieti, tak môže sfalšovať informácie o ID volajúcom prostredníctvom siete poskytovateľa služieb. Tento typ útoku je využívaný v systémoch, ktoré na identifikáciu užívateľov používajú informácie o identifikácii volajúceho. Príkladom môže byť, že niektoré spoločnosti poskytujúce mobilné telefóny používajú ID volajúceho na autentifikáciu účastníkov do ich hlasových schránok a v prípade, že sa ID volajúceho zhoduje s hlasovou schránkou užívateľa, nebude poskytovateľ od volajúceho užívateľa vyžadovať heslo pretože predpokladá, že hovor pochádza z mobilného telefónu užívateľa.

2.4 Nepovolený prístup

Nepovolený prístup je podobný ako podvody maskovania a odcudzenia identity. Rozdiel je však v tom, že útočník sa nemusí vydávať za iného užívateľa alebo sieťový prvok, ale skôr môže získať priamy prístup pomocou zraniteľnosti ako je preplnenie vyrovnávacej pamäte, konfigurácie, slabej signalizácie, alebo riadenie prístupu do siete. Útočník môže mať napríklad administratívny prístup na serveri SIP proxy, čím môže narušiť signalizáciu VoIP vymazaním súborového systému operačného systému a tým poškodiť hostiteľa a služby. Útočník tiež môže mať napríklad prístup na mediálnu bránu, kde nainštaluje škodlivý softvér na zhromažďovanie mediálnych paketov, a tiež k pasívnemu odpočúvaniu komunikácií predplatiteľa. Medzi najčastejšie typy útokov patrí napríklad:

- útok typu Man-in-the-Middle,
- útok odcudzenia identity,
- útok typu úplný kompromis.

Útoky typu *odcudzenie identity* zahŕňajú krádež alebo hádanie, ako napríklad hádanie kombinácie užívateľského mena a hesla. V prípade, že tieto údaje útočník uhádne, vydáva sa za užívateľa.

Pri útoku typu *úplná kompromisia* má útočník úplnú kontrolu nad systémom a môže vykonávať akékoľvek služby, príkazy a procesy v mene užívateľa. Príkladom tohto typu útoku môže byť útok červov, v ktorej červík beží na počítači obete a vydáva sa za neho pri nových komunikačných reláciách.

Útok typu *Man-in-the-Middle* (MITM) je možné považovať za jeden z najzávažnejších hrozieb pre bezpečnosť a dôveru existujúcich protokolov a systémov VoIP. Útočníci môžu pomocou signalizácie VoIP odpočúvať, presmerovať alebo ukradnúť vybrané hovory. Tento typ útoku používa techniku spoofingu DNS alebo otravu medzipamäte ARP, pomocou ktorej sa útočník môže dostať medzi SIP server a užívateľský SIP agent [7].

2.5 Podvod

Pri útokoch typu ***podvod*** sa útočník snaží zneužiť služby VoIP za účelom osobného alebo peňažného zisku. Tento typ útoku patrí k jednému z najkritickejších útokov pre telekomunikačných operátorov a poskytovateľov. Podvod je možné realizovať manipuláciou so signalizačnými správami alebo konfiguráciou VoIP komponentov [3]. Typickým príkladom môže byť podvod prostredníctvom

manipulácie s tokom hovorov, v ktorom zraniteľnosť využíva výhody ako sú SIP signalizačné správy spracované SIP proxy, pričom SIP proxy zvyčajne považuje za reláciu komunikáciu medzi dvoma užívateľmi po ukončení Three-Way-Handshake (INVATE, OK, ACK správy).

Do typu podvodného útoku môžeme tiež zaradiť *Phising*, kedy útočník ukradne užívateľom identitu, pričom sa tradične zameriava na užívateľov e-mailov, alebo na útoky, pri ktorých útočník využíva sfalšované legitímne finančné webové stránky, ako napríklad PayPal, eBay, ABC Banka a podobne. Obete sú tak zvyčajne nalákané na návštevu falošnej stránky a na poskytnutie dôležitých informácií, ako sú napríklad heslo, rodné priezvisko matky, číslo kreditných kariet, číslo sociálneho zabezpečenia a podobne [8].

Patria sem medzinárodné podvody so zdieľaným výnosom, nazývané *Mýtne podvody (Toll Fraud)*. V tomto type útoku môže podvodník vstúpiť do telefónneho systému a uskutočňovať podvodné diaľkové hovory z účtu obetí, alebo vykonávať volanie na čísla s prémiovou sadzbou. Patrí sem napríklad medzinárodné zdieľanie výnosov, použitie nelegálnych telefónnych kariet alebo zneužitie hovorov [53].

Ako ďalší príklad podvodného útoku môžeme uviesť útok typu *Wangiri*, ktorý používa metódu generovania krátkych telefónnych hovorov, kedy telefón obete zazvoní len raz a útočník následne zavesí, čím zanechá zmeškaný hovor v mobilnom telefóne napadnutej obete. Cieľom útoku je, aby obeť reagovala na zmeškaný hovor a zavolala naspäť na zmeškané telefónne číslo. Spätné volanie je potom presmerované na medzinárodné číslo s prémiovou sadzbou, alebo na iné číslo s poplatkom, na ktorom sa odporúča čakať na linke. Útočník takýto spoplatnený hovor obete zneužije ako príjem. [54].

3. Systém prevencie narušenia (IPS)

Systém prevencie narušenia (IPS) detekuje alebo zabraňuje pokusom o zneužitie slabín v zraniteľných systémoch alebo aplikáciách. Jeho úlohou je snaha identifikovať potenciálne hrozby na základe monitorovacích funkcií chráneného hostiteľa alebo siete a môže používať napríklad podpisové alebo detekčné metódy. IPS vykonáva kroky na zablokovanie všetkého, o čom sa domnieva, že predstavuje hrozbu pre chránený systém alebo nápravu identifikovanej hrozby, čím pomáha predchádzať narušeniu služby. Tento systém je ideálny pre prostredie, kde by akýkoľvek zásah mohol spôsobiť značné škody, ako sú napríklad databázy obsahujúce citlivé údaje [9].

Systémy prevencie narušenia môžu vykonávať aj komplikovanejšie pozorovania a analýzy, ako napríklad sledovanie a reagovanie na podozrivé vzory prenosu alebo paketov. Detekčné mechanizmy môžu zahŕňať:

- zhodu adries,
- generické porovnávanie vzorov,
- analýzu pripojenia TCP,
- detekciu anomálie paketov,
- zhodu portov TCP/UDP [10].

Systém IPS zaznamenáva informácie týkajúce sa pozorovaných udalostí, upozorňuje správcov bezpečnosti a vytvára správy. Pre zabezpečenie siete môže IPS systém automaticky prijímať preventívne a bezpečnostné aktualizácie, aby mohol nepretržite monitorovať a blokovať vznikajúce Internetové hrozby.

3.1 Druhy IPS

Medzi dva základné typy systémov prevencie narušenia IPS patria hostiteľské systémy na prevenciu vniknutia (Host IPS) a systémy prevencie narušenia siete (Network IPS). Okrem týchto typov sem zaraďujeme ešte dva ďalšie, a to analýzu správania siete (NBA) a bezdrôtové systémy na prevenciu narušenia (WIPS).

Host IPS (HIPS) je typ hostiteľskej prevencie narušenia systémov, ktorý sa okrem zistenia toho, že došlo k podozrivej udalosti snaží aj zastaviť činnosť tejto podozrivej aktivity. Rovnako ako systémy prevencie narušenia siete môžu využívať prístupy založené na *podpisoch* alebo *správaní*. Napríklad v prípade, ak sa útočník snaží preplniť medzipamäť tak, aby jeho škodlivý kód mohol bežať v pamäťovom priestore jadra, systém HIPS skontroluje systémové hovory a porovná ich buď so záznamom podpisov, alebo so zoznamom známych správ. V prípade ak HIPS identifikuje hovor ako škodlivý, neumožní mu prístup.

Existujú rôzne riešenia HIPS, ktoré využívajú rôzne prístupy. Väčšina z nich využíva agentov, ktorí sú centrálne riadení v systémoch využívajúcich ochranu. Agenti skúmajú systémové a API hovory aby zistili, kedy sa útočníci pokúšajú o hovor.

Pri prichádzajúcich hovoroch systémy HIPS, založené na *metóde detekcie na základe podpisu,* skontrolujú nelegálny zoznam hovorov, ktoré boli identifikované pri určitých druhoch útokov. V prípade, že prichádzajúci hovor obsahuje jeden z identifikovaných vzorov, neumožňuje mu prístup.

HIPS, ktoré sú *založené na type správania* obsahujú zvyčajne moduly pre jednotlivé služby systému API. Napríklad modul, ktorý kontroluje požiadavky medzi procesmi a súborovým systémom alebo modul, ktorý kontroluje požiadavky na register a mnoho ďalších.

Riešenia HIPS môžu chrániť pred mnohými typmi útokov, ako napríklad:

- zabrániť prístup k zoznamu kontaktov e-mailových klientov tak, aby sa útoky typu vírusy a červy nemohli šíriť ďalej týmto spôsobom,
- zabrániť zneužitiu privilégií, pri ktorom sa užívateľský účet pokúša získať prístup správcu alebo prístup root,
- zabrániť načítaniu napríklad Trojského koníka,
- zabrániť zmene systémových súborov, nastavenia registra a užívateľských účtov,
- zabrániť zneužitiu preplnenia medzipamäte [11].

Network IPS (NIPS) je typ sieťovej prevencie narušenia systémov, používaný na monitorovanie siete a na ochranu jej dôvernosti, integrity a dostupnosti. Všetky zariadenia NIPS umožňujú správcom definovať počítače, porty a aplikácie, ktoré je potrebné chrániť. Zdrojové a cieľové IP adresy a čísla portov sa používajú pre každý počítač a každú službu poskytovanú počítačom. Systém NIPS monitoruje sieť kvôli škodlivej aktivite alebo podozrivému prenosu analýzou aktivity protokolu. Inštalácia NIPS v sieti slúži pre vytvorenie zón fyzického zabezpečenia, čím robí sieť inteligentnú a rýchlo rozpozná dobrý prenos od zlého. Vyhradené zariadenia NIPS nemajú MAC adresu ani IP adresu, takže útočníci na tieto zariadenia nemôžu priamo zaútočiť. Medzi základné typy NIPS patria metódy *založené na rýchlosti* a metódy založené *na základe obsahu.*

Systémy NIPS *založené na rýchlosti* používajú prahové hodnoty, ktoré zisťujú, či do siete prichádza príliš veľa pripojení, chýb alebo paketov. Spôsob, ktorým riešia ochranu založenú na rýchlosti je odlišný, všetky zariadenia ale umožňujú správcom definovať počítače, porty a aplikácie, ktoré je potrebné chrániť.

Zariadenia NIPS, ktoré sú založené na základe *metódy obsahu* vyhľadávajú anomálne správanie a anomálie protokolu, aby odhalili škodlivé užitočné zaťaženie. Medzi jeho typický príklad patrí napríklad prenos FTP smerujúci na port 53, binárny kód v rámci hesla užívateľa, alebo nadmerný počet bitov prichádzajúcich z webového prehľadávača a mnoho ďalších.

Medzi potenciálne škodlivé úpravy sieťových a transportných hlavičiek paketu je možné zaradiť napríklad:

- nesprávnu dĺžku hlavičky alebo poľa,
- poškodené kontrolné súčty,
- nesprávne prekrývanie segmentácie TCP,
- nekonzistentné použitie príznakov v poliach hlavičiek.

Medzi anomálie aplikačnej vrstvy je možné zaradiť napríklad:

- nezákonné použitie príkazu protokolu,
- neobvykle dlhé alebo krátke dĺžky poľa, ktoré by mohli naznačovať preplnenie medzipamäte,
- nesprávne hodnoty polí a ich kombinácie,
- a iné [12].

Analýza správania siete (NBA) je analytická technika založená na skúmaní sieťového prenosu pre detekovanie nežiaduceho správania a identifikovania hrozieb, čím dokáže zvýšiť sieťovú bezpečnosť. Medzi jej základné metódy detekcie patrí *detekcia na základe anomálií a detekcia analýzy stavového protokolu.*

Detekcia na základe anomálií je metóda, hľadajúca odchýlky od toho, čo je pre nás známe ako ,,normálne“ správanie v systémovej alebo sieťovej aktivite. Po jej spustení, za určité tréningové obdobie zostaví profil toho, čo považuje za normálne správanie, pričom nezrovnalosti vyhodnocuje ako škodlivé. Tento typ detekcie je vynikajúci pri identifikácii nových hrozieb. Avšak v prípade, že dôjde k narušeniu tréningového obdobia, škodlivé správanie môže byť zaznamenané ako ,,normálne“ a nevyhodnotí detekciu sieťového prenosu správne.

Detekcia analýzy stavového protokolu je podobná detekcii na základe anomálie, pretože tiež vyhľadáva odchýlky od ,,normálneho“ správania siete alebo systému. Vyhodnocuje správanie na základe univerzálnych profilov.

Systém NBA by sa mal používať ako rozšírenie NIPS alebo IDS IPS, aby bola zabezpečená vrstevná ochrana.

Bezdrôtové systémy na prevenciu narušenia (WIPS) analyzujú rádiové spektrum v bezdrôtovej sieti na detekciu a hlásenie vniknutia, porušenie sieťových zásad a neoprávneného použitia. WIPS môžeme implementovať troma spôsobmi, a to *monitorovaním prekrytia*, *integrovaným monitorovaním* a *hybridným monitorovaním.*

Monitorovanie prekrytia, v ktorom sú bezdrôtové senzory umiestnené v celej fyzickej sfére siete. *Integrované monitorovanie*, ktoré na zabezpečenie a pripojenie namiesto bezdrôtových senzorov využíva konzoly prístupového bodu. *Hybridné monitorovanie* používa na monitorovanie a pripojenie senzory a AP konzoly.

Bezdrôtové systémy na prevenciu narušenia môžu tiež zhromažďovať informácie o zariadeniach pripojených k sieti, čo je veľmi efektívne pri detekcii a pri predchádzaní útokov ako napríklad DoS útok, neoprávnený prístup a iné. Monitorovanie prekrytia a integrované monitorovanie majú jedinečné obmedzenia, preto väčšina organizácií používa hybridné monitorovanie [16].

3.2 Metódy detekcie

Väčšina systémov prevencie narušenia využíva jednu z troch detekčných metód, a teda detekciu na základe podpisu, štatistickú detekciu na základe anomálie alebo detekciu analýzy stavového protokolu.

Detekcia na základe podpisu – na základe podpisu systém detekcie vniknutia (IDS) monitoruje pakety v sieti a porovnáva ich s vopred určenými vzormi útokov, známymi ako ,,podpisy". Po odhalení zneužitia sa podpis uloží a zaznamená do neustále rastúceho slovníka podpisov.

Detekcia podpisu pre systém prevencie narušenia sa delí na dva typy:

Prvý typ zahŕňa *podpisy zamerané na zneužitie*, ktoré identifikujú jednotlivé zneužitia spustením jedinečných vzorov konkrétneho pokusu o zneužitie. Systémy na prevenciu narušenia dokážu identifikovať zneužitie nájdením zhody s podpisom smerujúcim k zneužitiu v toku prenose.

Druhým typom sú *podpisy zamerané na zraniteľnosti* v systéme. Sú to širšie podpisy, zameriavané na základnú zraniteľnosť v systéme. Tieto podpisy umožňujú ochranu sietí pred variantami zneužitia, ktoré sa priamo nemusia pozorovať, ale len zvyšujú riziko falošných poplachov.

Štatistická detekcia na základe anomálie – systém detekcie vniknutia monitoruje sieťový prenos a porovnáva ho s očakávanými vzormi prenosu. Na základe vyhodnotenia určí, čo je pre sieť ,,normálne" (aký druh paketov a protokolov všeobecne používajú cez sieť). V prípade, že nie sú podmienky prenosu správne nakonfigurované, môže falošne vyvolať poplach.

Detekcia analýzy stavového protokolu – táto metóda identifikuje odchýlky protokolu porovnávaním pozorovaných udalostí s vopred stanovenými profilmi aktivity normálnej aktivity [10].

3.3 Architektúra IPS

Architektúra systémov prevencie narušenia je jedna z najdôležitejších aspektov. Môžeme ju považovať za efektívnu v prípade, že každý stroj, zariadenie, komponenty alebo proces vykonáva svoju úlohu efektívne a často koordinovaným spôsobom, čím docielime efektívne spracovanie a výstup informácií. Zle navrhnutá architektúra môže spôsobiť nežiadúce dôsledky, ako napríklad, že údaje nie sú k dispozícii v prípade, že je to potrebné, alebo môže vyvolať spomalenie siete, či nedostatok vhodných a včasných reakcií. Architektúru môžeme rozdeliť na tri typy:

- jednoúrovňová architektúra,
- viacúrovňová architektúra,
- Peer-to-Peer architektúra.

Jednoúrovňová architektúra patrí k jednej z najzákladnejších architektúr, v ktorej komponenty systémov detekcie a prevencie narušenia zhromažďujú a spracovávajú údaje samostatne bez toho, aby odovzdávali výstup, ktorý zhromažďujú do inej sady komponentov. Typickým príkladom jednoúrovňovej architektúry je hostiteľský nástroj na detekciu narušenia, ktorý berie výstup

systémových protokolov a porovnáva ho so známymi vzormi útoku. Medzi výhody jednoúrovňovej architektúry patrí jednoduchosť, nízka cena a nezávislosť od ostatných komponentov.

Viacúrovňová architektúra zahŕňa viac komponentov, ktoré si navzájom odovzdávajú informácie. Mnoho dnešných systémov detekcie narušenia sa skladá napríklad z troch základných komponentov, ako sú senzory, analyzátory alebo agenti a manažéri. Senzory vykonávajú zber informácií, zhromažďujú údaje zo systémových protokolov a iných zdrojov, ako napríklad osobných brán firewallu. Najčastejšie sa používajú sieťové snímače. Senzory odovzdávajú informácie agentom, niekedy tiež označovaným ako analyzátory, ktorí monitorujú rušivú aktivitu na ich jednotlivých hostiteľoch. Agenti sa zvyčajne špecializujú na výkon jednej funkcie. V prípade, že agent zistí, že došlo k útoku, odošle informácie manažérovi, ktorý vykoná napríklad zhromažďovanie a zobrazovanie upozornení na konzole, spustenie pageru alebo zavolanie na číslo mobilného telefónu a podobne. Medzi jeho výhody patrí hlavne vyššia efektívnosť a hĺbka analýzy.

Peer-to-Peer architektúra zahŕňa výmenu informácii o detekcii a prevencii narušenia medzi rovnocennými komponentami, z ktorých každá vykonáva rovnaké druhy informácií. Architektúru typu Peer-to-Peer najčastejšie používajú spolupracujúce brány firewallu, ktoré v prípade, že získajú informácie o udalostiach vyskytujúcich sa v systéme, odovzdajú ich ďalšiemu, čím môžu spôsobiť zmenu informácií v zozname a tiež druhý firewall môže odoslať informácie, ktoré spôsobujú zmeny v prvom a môžu zmeniť jeho správanie. Hlavnou výhodou tejto architektúry je jej jednoduchosť [18].

3.4 Nástroje IPS

Z hľadiska analýzy súčasných nástrojov používaných pre realizáciu IPS systému existuje niekoľko voľne dostupných alebo platených nástrojov, ktoré sú používané na prevenciu a detekciu systému narušenia. Medzi najčastejšie voľne dostupné nástroje patrí napríklad nástroj Snort, Bro-IDS (Zeek), Suricata alebo SolarWinds Security Event Manager [36].

Nástroj Snort je open source sieťový IPS a IDS systém, ktorý vykonáva analýzu v reálnom čase a generuje výstrahy pri detekcii hrozieb v sieťach IP. Okrem toho môže tiež vykonávať analýzu protokolov, vyhľadávanie a porovnávanie obsahu, alebo zisťuje rôzne útoky. V režime systému detekcie narušenia kontroluje informácie o prenose a hlási výstrahy, ale nezabraňuje možným útokom. V režime systému prevencie narušenia podniká kroky na zabránenie útokom [37]. Medzi hlavné funkcie, ktoré vykonáva patrí monitorovanie sieťového prenosu a analýza na základe definovanej sady pravidiel, ďalej vykonávanie klasifikácie útokov a vyvolávanie akcie proti zhodným pravidlám [38].

Nástroj Snort môže pracovať v troch rôznych režimoch. Prvý režim je *Sniffer paketov*, v ktorých nástroj Snort číta IP pakety a zobrazuje ich na konzole. Druhý režim, nazývaný *Logger paketov*, v ktorom sa zaznamenávajú protokoly IP a posledný, tretí režim *Systém detekcie narušenia,* ktorý využíva sady pravidiel pre kontrolu IP paketov [39].

Nástroj Bro-IDS (Zeek) bol v roku 2018 premenovaný na nástroj Zeek. V minulosti bol označovaný ako Bro-IDS, a dnes ho môžeme poznať pod názvom Zeek-IDS. Tento nástroj je určený na zisťovanie anomálie správania v sieti na účely kybernetickej bezpečnosti. Predovšetkým funguje ako bezpečnostný monitor, ktorý podrobne kontroluje komunikáciu a zisťuje, či neobsahuje príznaky podozrivej činnosti. Hlavnou výhodou tohto nástroja sú rozsiahle sady protokolových súborov,

zaznamenávajúcich komplexný záznam každého spojenia alebo prepisy aplikačnej vrstvy. Okrem iného poskytuje tiež zabudované funkcie pre celý rad analytických a detekčných úloh, vrátane extrakcie súborov, detekcie škodlivého malvéru, hlásenie zraniteľných verzií softvéru nachádzajúcich sa v sieti, alebo overovanie reťazcov certifikátov SSL [40].

Architektúru nástroja Zeek možno rozdeliť do dvoch hlavných zložiek. Prvou zložkou je *Event Engine*, slúžiaca k redukcii prichádzajúcich paketov a druhou zložkou je *Policy Script Interpreter,* vykonávajúca sadu obslužných rutín napísaných v skriptovacom jazyku Zeek [41].

Nástroj Suricata je otvorený, bezplatný a rýchly detekčný modul, ktorý je schopný pracovať v režime IDS a IPS. Pretože nástroj Snort patrí k najpopulárnejším a najbežnejšie používaným IDS nástrojom, nový nástroj Suricata vznikol ako alternatíva ku komunikačnému nástroju Snort, ktorý obsahoval výkonnostné obmedzenia architektúry s jedným vláknom. Architektúra Suricaty je implementovaná ako viacvláknová, čím sa zvýši schopnosť prijímania paketov a nedôjde tak k ignorovaniu prichádzajúcich paketov z dôvodu obmedzenej kapacity [42]. Podrobnejší popis nástroja Suricata je uvedený v podkapitole 4.7.

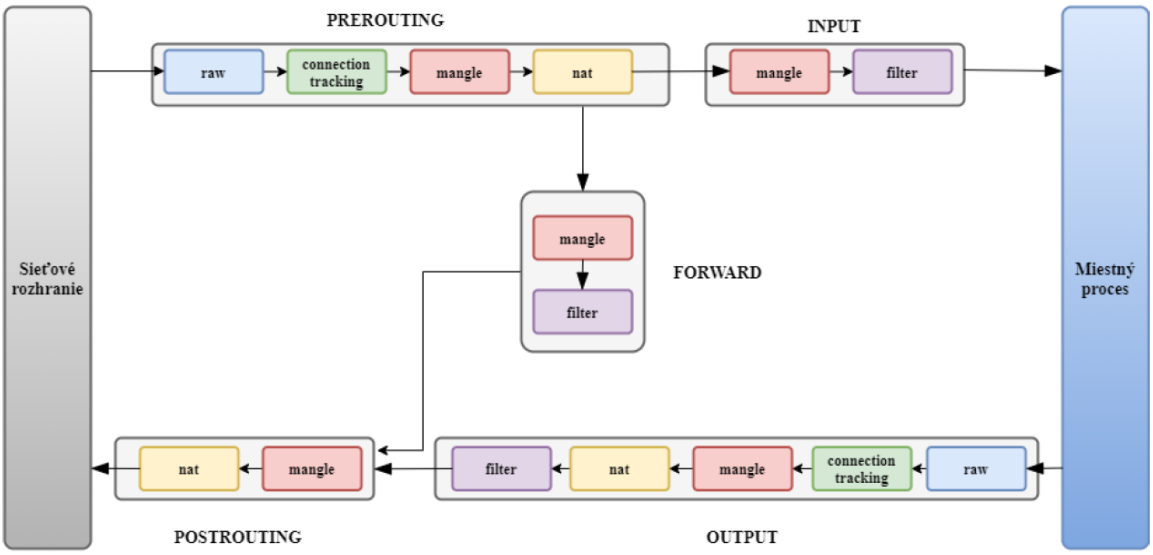
Nástroj SolarWinds Security Event Manager je virtuálne zariadenie na správu informácií a udalostí, ktoré zvyšuje hodnotu existujúcich bezpečnostných produktov a tiež zvyšuje efektivitu pri správe a monitorovaní siete. Security Event Manager (SEM) uľahčuje neustále sledovanie, preposielanie alebo archiváciu protokolových súborov a obsahuje zabudované šifrovanie prepravy a úložiska. Jedná sa o platenú možnosť detekcie narušenia a prevencie systému, existuje však tiež možnosť vyskúšať si tento nástroj zadarmo po dobu tridsať dní [43].

Pri realizácii testovania v rámci diplomovej práce som si vybrala ako metódu detekcie nástroj Suricata, ktorý pracuje v režime prevencie systému. Testovaná bude s pobočkovou ústredňou Asterisk, na ktorú sú nainštalované dva softvérové telefóny PhonerLite a Jitsi, pričom bude plniť funkciu prevencie systému a na základe vytvorených pravidiel sa bude snažiť o blokovanie podozrivej komunikácie.

4. IPtables, NFqueues a Suricata

Filtračný nástroj IPtables je softvérové riešenie, umožňujúce správcovi systému konfigurovať low-level paketový filter – Netfilter, ktorý je súčasťou Linuxového jadra. Pomocou nástroja IPtables definujeme pravidlá a príkazy, ktoré uľahčujú preklad sieťových adries, filtrovanie paketov a manipuláciu s paketmi. Medzi najčastejšie pravidlá, ktoré môžeme definovať pomocou IPtables patria povolenia alebo blokovania rôznych služieb podľa portu, sieťového rozhrania a zdrojovej adresy IP. IPtables a Netfilter sú nástupcami IPchains a IPfwadm v starších verziách Linuxu. Mechanizmus filtrovania paketov je organizovaný do troch rôznych štruktúr:

- *tabuľky*, sú súbory, ktoré sa spájajú s podobnými akciami. Tabuľky sa skladajú z niekoľkých reťazcov.
- *Reťazce*, predstavujú reťazce pravidiel, kde po prijatí paketu IPtables vyhľadá príslušnú tabuľku a potom ju spustí v rámci reťazca pravidiel, pokiaľ nenájde zhodu.
- *Ciele*, ktoré rozhodujú, čo majú urobiť s paketom. Spravidla je to prijať, vylúčiť alebo odmietnuť.



Obrázok 3 Základná štruktúra systému IPtables

Sieťový prenos je tvorený paketmi, pričom systém IPtables identifikuje prijaté pakety a pomocou súboru pravidiel rozhoduje, čo urobí s paketmi. Po príchode paketu na rozhranie IPtables porovná paket s pravidlami v reťazcoch v zásade po jednom a keď nájde zhodu, tak vykoná akciu. V prípade, že nenájde zhodu so žiadnym z pravidiel, vykoná to, čo hovorí predvolené pravidlo.

4.1 Tabuľky

V linuxových distribúciách má filtračný systém IPtables štyri predvolené tabuľky, a to tabuľky Filter, NAT (Network Address Translation), RAW a MANGLE. Reťazce umožňujú správcovi riadiť, kde na doručovacej ceste paketu bude pravidlo vyhodnotené. Pretože každá tabuľka má viac reťazcov, vplyv tabuľky je možné uplatniť vo viacerých bodoch spracovania.

Tabuľka Filter – patrí k najčastejšie používanej tabuľke. Rozhoduje o tom, či nechať paket pokračovať na určené miesto alebo odmietnuť jeho žiadosť. Táto činnosť sa nazýva ,,filtrovanie paketov". Obsahuje predvolené reťazce ako Input, Output a Forward.

Input – tento reťazec spracováva všetky pakety, ktoré sú určené na náš server, a tiež riadi správanie prichádzajúcich pripojení.

Forward – tento typ reťazca je využívaný pre pakety smerované cez systém. Môžeme si predstaviť smerovač, dáta sú do neho vždy odoslané, ale zriedka sú skutočne určené pre samotný smerovač, údaje sa preposielajú k jeho cieľu.

Output – reťazec obsahuje pravidlá pre pakety generované lokálne.

Tabuľka NAT – obsahuje pravidlá prekladu sieťových adries pre smerovanie paketov do sietí, ku ktorým nie je možné získať priamy prístup. Používame ju teda v prípade, že chceme zmeniť cieľ alebo zdroj paketu. Obsahuje reťazce ako Prerouting, Output a Postrouting.

Postrouting - je určené pre pakety opúšťajúce sieťové rozhranie.

Prerouting –je uplatnené na pakety v momente, keď pakety dorazia na rozhranie.

Output – funguje rovnako ako v prípade tabuľky Filter.

Tabuľka RAW – má veľmi úzko definovanú funkciu, jeho jediným účelom je poskytnúť mechanizmus na označovanie paketov za účelom sledovania spojenia. Obsahuje reťazce ako Prerouting a Output.

Prerouting – pre zmenu prichádzajúcich pripojení.

Output – pre zmenu lokálne generovaných paketov.

Tabuľka MANGLE – upravuje vlastnosti hlavičky IP paketov, pričom môžeme napríklad upraviť hodnotu Time to Live (TTL) paketu, a to buď predĺžením alebo skrátením počtu platných sieťových skokov. Podobne môžeme zmeniť aj ďalšie hlavičky. Obsahuje reťazce ako Prerouting, Postrouting, Output a Input.

Input – pre prichádzajúce pakety.

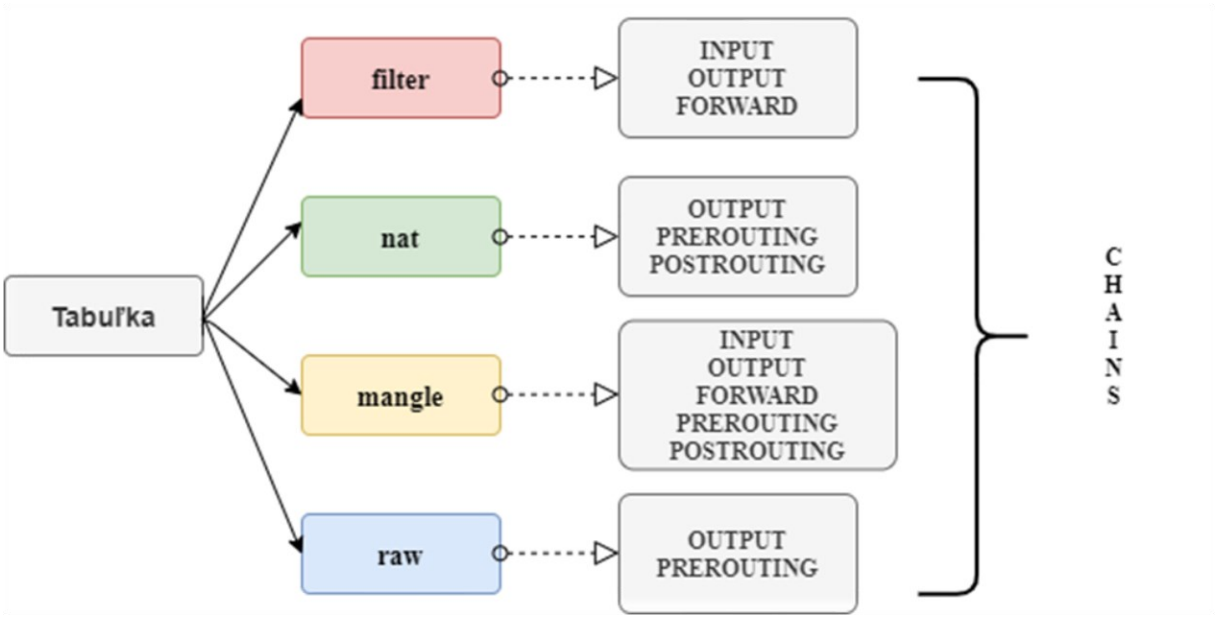
Output – pre zmenu lokálne generovaných paketov.

Forward– pre smerovanie paketov.

Prerouting – pre zmenu prichádzajúcich pripojení.

Postrouting – na varovanie odchádzajúcich pripojení.

Niektoré verzie systému Linux používajú tiež tabuľku ***Zabezpečenie*** pre správu špeciálnych pravidiel. Obsahuje reťazce ako Input, Output a Forward, podobne ako tabuľka Filtra [13].



Obrázok 4 Štruktúra tabuliek pre IPtables

4.2 Ciele

Cieľ je akcia, ktorá sa spustí v prípade, že paket spĺňa kritéria zhody pravidla. Ciele všeobecne môžeme rozdeliť do dvoch kategórií:

Ukončovacie ciele vykonávajúce činnosť, ktorá ukončí hodnotenie v reťazci a vráti kontrolu sieťovému filtru. V závislosti od poskytnutej návratovej hodnoty môže paket vylúčiť alebo mu umožniť pokračovať v ďalšej fáze spracovania.

Nekončiace ciele, ktoré vykonávajú činnosť a pokračujú v hodnotení v rámci reťazca. Aj keď každý reťazec musí nakoniec odovzdať konečné rozhodnutie o ukončení, je možné vopred vykonať akýkoľvek počet nekončiacich cieľov. Dostupnosť každého cieľa v rámci pravidiel závisí od kontextu a môžu napríklad určovať typ tabuľky a reťazca.

Medzi základné ciele filtračného systému IPtables patrí:

Accept – pravidlo akceptuje pakety prichádzajúce cez bránu IPtables firewallu.
Drop – v tomto pravidle sa paket zahodí a jeho odosielateľ nedostane informáciu o tom, čo sa sním stalo.
Return – pravidlo odošle paket späť do pôvodného reťazca tak, aby ho mohol porovnať s inými pravidlami.
Reject – brána firewallu IPtables odmietne paket a odošle chybu pripojovaciemu zariadeniu [14].

4.3 Moduly

Systém IPtables okrem toho, že pracuje na základe IP adresy alebo portu, dokáže svoju funkcionalitu rozšíriť o relatívne veľké množstvo funkcií. Spojenia sledované systémom môžu byť rozdelené do nasledujúcich stavov:

NEW – tento stav reprezentuje prvý paket, ktorý nie je spojený s existujúcim pripojením. Do systému s týmto označením bude pridané nové pripojenie. Tento stav nastáva v prípade protokolov založených na pripojení, ako sú protokoly TCP a UDP.

ESTABLISHED – tento stav nastáva zo stavu NEW, keď dostane platnú odpoveď v opačnom smere, a teda platí, že pakety sú už súčasťou existujúceho spojenia. Pre TCP pripojenie je to správa SYN/ACK a pre UDP a ICMP prenos stačí odpoveď, kde sa prepne zdroj a cieľ pôvodného paketu.

RELATED – je stav, ktorý je už naviazaný na existujúce pripojenie zo stavu ESTABLISHED. Príkladom môže byť prenos údajov FTP, alebo to môžu byť pokusy o pripojenie inými protokolmi.

INVALID – pakety v tomto stave môžeme označiť za neplatné, ak nie sú spojené s existujúcim pripojením a nie sú vhodné na vytvorenie nového spojenia, ak ich nemôžeme identifikovať alebo zmeniť z iných dôvodov.

UNTRACKED - pakety môžu byť označené ako nesledované, ak boli vyňaté zo sledovania pripojenia v RAW tabuľke s cieľom vyhnúť sa sledovaniu.

DNAT – virtuálny stav používaný pre pakety, ktorých cieľová IP adresa bola zmenená operáciami v NAT tabuľke.

SNAT - virtuálny stav používaný pre pakety, ktorých zdrojová IP adresa bola zmenená operáciami v NAT tabuľke [15].

Stavy sledované v systéme sledovania pripojení umožňujú správcom vytvárať pravidlá, zameriavajúce sa na konkrétne body počas životnosti spojenia, pričom poskytuje funkcie potrebné pre dôkladnejšie a bezpečnejšie pravidlá.

4.4 Príkazy a pravidlá

Príkazy slúžia pre definovanie konkrétnej akcie, ktorá má byť vykonaná. V príkazovom riadku môže byť zadaný len jeden príkaz, pokiaľ nie je uvedené inak. Pre dlhé názvy príkazov môžeme použiť len toľko písmen príkazov tak, aby sa písmená príkazov pre filtrovanie sieťového prenosu pomocou IPtables odlíšili od ostatných.

- *A - - append chain rule-specification:* pridať špecifikáciu pravidla reťazca.
- *D - - delete chain rule-specification:* odstráni jeden alebo viac pravidiel z vybraného reťazca.

- *I - - delete chain rule-specification:* umožní vložiť jedno alebo viac pravidiel do vybraného reťazca.
- *R - - replace chain rulenum rule-specification:* nahradí pravidlo vo vybranom reťazci.
- *L - - list:* vypíše pravidlá zvolenej triedy, a ak nie je vybraný žiaden reťazec, tak vypíše všetky pravidlá.
- *F - - flush:* umožní zmazať pravidlá vo vybranej triede.
- *Z - - zero:* vynuluje počítadlo paketov a bajtov vo všetkých reťazcoch.
- *N – new-chain:* vytvorí novú užívateľskú triedu.
- *X - - delete-chain:* odstráni užívateľom zadaný reťazec.
- *P - - policy:* nastaví politiku reťazca na daný cieľ.
- *E - - rename-chains:* premenuje reťazec zadaný užívateľom.

Nasledujúce parametre tvoria špecifikáciu pravidiel, ako napríklad:

- *p - - protocol*: definuje protokol, ku ktorému sa pravidlo vzťahuje. Protokoly môžu byť TCP,UDP,ICMP alebo iné.
- *s - - source*: adresa/maska, ktorá nás oboznamuje o tom, aká zdrojová adresa vyhovuje aktuálnemu pravidlu.
- *d - - destination*: adresa/maska, ktorá nás oboznamuje o tom, aká cieľová adresa vyhovuje aktuálnemu pravidlu.
- *j - - jump target*: určuje pravidlá čo robiť v prípade, ak sa paket zhoduje.
- *g - - goto chain*: v prípade, že dáta splnia podmienku, tak budú pokračovať v zadanom reťazci užívateľa.
- *i - - in-interface*: definuje, cez ktoré rozhranie bol paket prijatý. Sú to pakety reťazcov Input, Forward a Prerouting.
- *o - - out-interface*: definuje rozhranie, cez ktoré bude paket odosielaný. Sú to pakety reťazcov Forward, Output a Postrouting.
- *f - - fragment*: udáva pravidlo až od druhého a ostatných fragmentových dát.
- *c - - set-counters*: umožňuje inicializáciu správy počítania paketov.
- *v - - verbose*: umožňuje príkazu zobraziť názov rozhrania, možnosti pravidla v prípade, že existujú a podobne.
- *n - - numeric*: je číselný výstup IP adresy a čísla portov. V predvolenom nastavení sa program pokúsi zobraziť ich ako názvy hostiteľov, názvy sietí alebo služby.
- *x - - exact*: zobrazuje presnú hodnotu počítadla paketov a bajtov.

4.5 NFtables

Podobne ako filtračný nástroj IPtables, tak aj jeho novšia verzia NFtables bol vyvinutý organizáciou Netfilter. NFtables je nástupca aplikácie IPtables a je bežne používaný pre tvorbu komplexnejších a rozsiahlejších súborov bezpečnostných pravidiel a firewallov. Jeho hlavnou úlohou je filtrovanie sieťového prenosu s cieľom jednoduchšieho a univerzálnejšieho riešenia, ktorý potrebuje menej kódov než nástroj IPtables [23]. Medzi jeho rozdiely v porovnaní s IPtables patria napríklad:

- ľahšie sa používa a kombinuje všetky nástroje rámca IPtables, ako napríklad iptables, ip6tables, arptables a iné do jedného nástroja *nft.*

- Tabuľky NFtables neobsahujú žiadne preddefinované predvolené tabuľky a reťazce, ako napríklad filter/NAT alebo FORWARD/INPUT.
- Systém NFtables má lepšiu a ľahšiu syntax v porovnaní s nástrojom IPtables, písanie kódov, rovnako ako jeho údržba, je jednoduchšia.
- Systém NFtables patrí k efektívnejším nástrojom, pretože je jednoduchý, urýchľuje konfiguráciu, a pakety môžu používať rôzne porty.
- Tvorba pravidiel je výrazne voľnejšia a jedno pravidlo môže obsahovať viac výrazov, alebo môžeme pre jedno pravidlo definovať viac cieľov.
- Firewall neobsahuje automatické počítadlo na reťazcoch a pravidlách [24].
- Zlepšuje výkon použitím dátových štruktúr, ktoré pomáhajú pri rýchlom vyhľadávaní v pamäti [25].

Systém NFtables ale zachováva niektoré časti filtračného systému IPtables, ako napríklad používanie existujúcej infraštruktúry, systém sledovania pripojenia, NAT, logovaciu infraštruktúru, fronty užívateľského priestoru a iné [23].

4.6 NFqueue

Nástroj *NFqueue* je jadro a modul užívateľského režimu pre správu sieťových paketov systému IPtables. Môžeme pomocou neho písať cieľové moduly Netfiltra v užívateľskom priestore, fungujúcom prostredníctvom ,,zásuviek“ Netlink, alebo pomocou jednoduchšej voľby predvolenej knižnice *ibnetfilter_queue.so.* Nástroj NFqueue používame okrem manipulácie a sledovania sieťového prenosu hlavne aj na filtrovanie alebo formovanie prenosu [17]. Ciele definované pre nástroj NFqueue sú voľne dostupné, avšak existujú aj v podobe komerčných systémoch prevencie narušenia. Podobne tomu je aj tak v prípade nástroja Suricata.

Cieľom nástroja NFqueue je rozšírenie cieľa Queue, ktorý nám na rozdiel od nástroja Queue umožňuje vložiť paket do akéhokoľvek konkrétneho radu, identifikovaným 16-bitovým číslom rady.

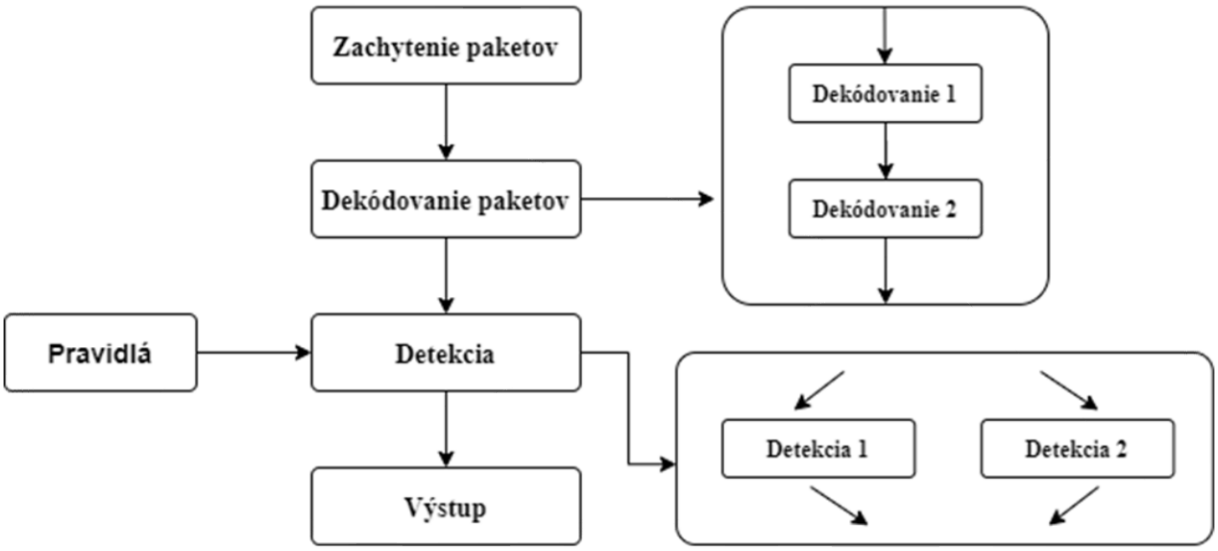
4.7 Nástroj Suricata

Nástroj Suricata je používaný na detekciu hrozieb v reálnom čase, chrániaci sieť pred hrozbami aktívnym sledovaním sieťového prenosu a detekciou škodlivého správania na základe napísaných pravidiel. Nástroj Suricata patrí k otvoreným systémom detekcie a prevencie narušenia siete, v ktorej je schopná identifikovať škodlivé správanie a zablokovať ho, čím eliminuje sieťové hrozby. Nástroj bol vyvinutý nadáciou Open Security Foundation (OISF) a od ostatných nástrojov sa líši predovšetkým v tom, že je viacvláknový, a jedna inštancia sa môže vykonávať pri oveľa vyššom objeme prenosu. Pre protokoly aplikačnej vrstvy je k dispozícii väčšia podpora, podporuje hašovanie a extrakciu súborov [20].

4.7.1 Architektúra

Architektúra nástroja Suricata je rozdelená na dve časti, na dekódovanie a na detekciu. ***Dekódovacie moduly*** pridávajú informácie do vnútornej reprezentácie paketov v tomto nástroji. Dekódovacie funkcie čítajú paket a ukladajú dekódované dáta do vnútornej reprezentácie paketov. Dekódovacie funkcie sa volajú po jednom pakete. ***Detekčné moduly*** sa spoliehajú na vnútorné

znázornenie a poskytujú kľúčové slová na použitie v pravidlách. Detekcia sa riadi pravidlami a závisí od kroku dekódovania. Pravidlá sa zhodujú s vnútorným zastúpením paketov a proces párovania je rozdelený do niekoľkých detekčných modulov, vo všetkých ktorých sa párovanie uskutočňuje. Na rozdiel od dekódovania je detekcia paralelná a jeden paket je možné spracovať vo viacerých detekčných moduloch súčasne. Architektúra nástroja Suricata je znázornená na obrázku 5 [21].



Obrázok 5 Architektúra nástroja Suricata

4.7.2 Štruktúra

Signatúry majú v nástroji Suricata veľmi dôležitú úlohu. Pravidlá pre signatúry sa skladajú z:

- *Akcií*, určujúcich čo sa stane, ak sa signatúra zhoduje.
- *Hlavičky*, definujúcej protokol, IP adresy, porty a smer pravidla.
- *Možnosti* pravidla, ktoré definujú špecifické pravidlá.

Akcia v nástroji Suricata definuje hranicu medzi systémom prevencie narušenia a systémom detekcie narušenia. Toto pravidlo obsahuje štyri typy akcií, ako sú *Pass, Drop, Reject* a *Alert,* ktoré rozhodujú o tom, čo sa stane s paketom.

Pass – ak sa systém zhoduje s podpismi/signatúrami, tak nástroj Suricata zastaví skenovanie paketu a prejde na koniec pravidla.

Drop – týka sa iba režimu IPS. V prípade, že systém nájde podpis/signatúru, ktorý sa zhoduje a obsahuje *drop*, okamžite zastaví a paket nebude ďalej odosielaný.

Reject – aktívne odmietnutie paketu. Odosielateľ a príjemca dostanú paket odmietnutia. Existujú dva typy paketov odmietnutia, ktoré sa automaticky vyberú. Prvý typ paketu sa týka protokolu TCP a nazýva sa resetovací paket. Všetky ostatné protokoly sa nazývajú pakety ICMP-error.

Alert – ak sa podpis/signatúra zhoduje a obsahuje výstrahu (alert), s paketom sa bude zaobchádzať ako s akýmkoľvek iným neohrozujúcim paketom, avšak nástroj Suricata vygeneruje výstrahu, ktorú si dokážu všimnúť len správcovia systému [19].

Protokol, informujúci v signatúre nástroj Suricata, ktorého protokolu sa to týka. Patria sem štyri základné protokoly medzi ktoré zaraďujeme TCP, UDP, ICMP a IP. Nástroj Suricata tiež môže použiť niektoré protokoly z aplikačnej vrstvy alebo siedmej vrstvy modelu OSI, medzi ktoré najčastejšie patrí napríklad HTTP, FTP, DNS, SSH a iné.

Zdrojová a cieľová IP adresa, pre ktoré môžeme použiť IPv4 a IPv6 kombinovane alebo aj oddelene.

Porty, pretože sieťový prenos prichádza a odchádza cez porty. Rôzne porty majú rôzne čísla portov, napríklad predvolený port HTTP má číslo 80, ale zvyčajne má port s číslom 443. Port preto neurčuje, ktorý protokol sa v komunikácii používa, ale určuje, ktorá aplikácia prijíma údaje.

Určovanie *smeru,* rozhodujúce o tom, akým spôsobom sa musí signatúra zhodovať. Najčastejšie sa používa šípka smerom doprava, ktorá znamená, že sa môžu pakety zhodovať iba v rovnakom smere.

Možnosti, sú uzavreté zátvorkami a oddelené bodkočiarkami. Môžu obsahovať správu alebo tiež rozličné kľúčové slová [21].

Príklad pravidla:

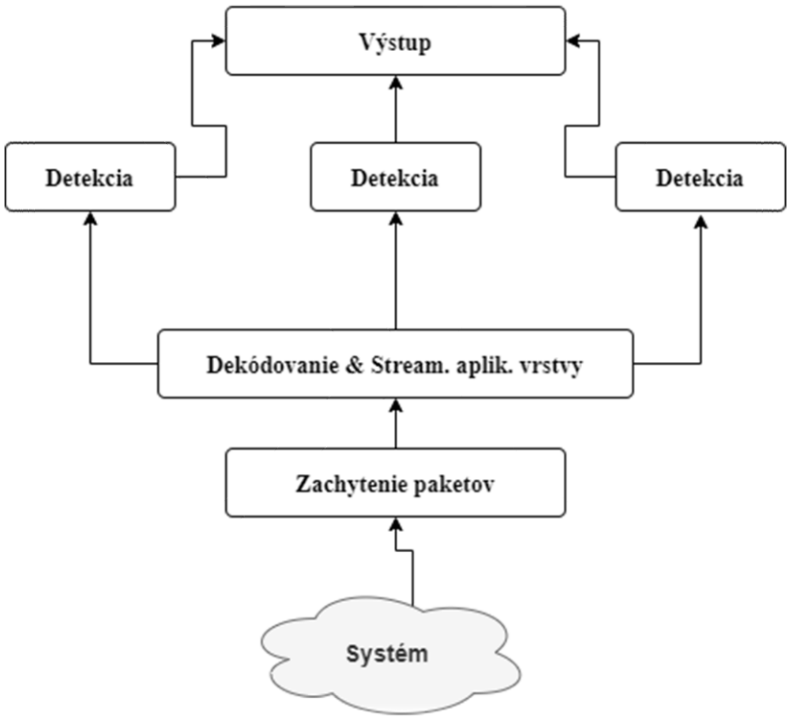
```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";
flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK "; pcre:"/NICK .*USA.*[0-9]
{3,}/i"; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124;
rev:2;)
```

Obrázok 6 Príklad tvorby pravidla nástroja Suricata

Pravidlo sa skladá z troch častí, pričom jeho prvá časť vyznačená ***červenou farbou*** označuje typ *akcie*. V tomto prípade definujeme akciu typu *drop*, pomocou ktorej chceme aby v prípade, že systém nájde signatúru, ktorá sa zhoduje a obsahuje *drop*, okamžite zastaví sieťový prenos a paket nebude ďalej odosielaný. Druhá časť pravidla je vyznačená ***zelenou farbou*** a predstavuje *hlavičku*, ktorá definuje typ *protokolu*, *rozhranie siete* a *smer*, ktorým sa pakety majú zhodovať. Tretia časť pravidla, vyznačená ***modrou farbou***, predstavuje vlastnosti prenášaných dát, napríklad pole *msg* predstavuje popis prenášanej správy, pole *flow* priraďuje smer toku pre naviazanie spojenia z klienta na sever, pole *flowbits* v prvej časti popisuje akciu, ktorú vykoná a v druhej časti názov flowbitu, pole *content* definuje názov metódy, *reference* obsahuje stránku týkajúcu sa pravidla, *classtype* poskytuje informácie o klasifikácii pravidiel a upozornení, *sid* dáva každej signatúre vlastné ID a posledné pole *rev* predstavuje verziu podpisu.

4.7.3 Funkcia IPS

Nástroj Suricata je možné spustiť aj ako systém detekcie narušenia systému a tiež ako aj systém prevencie narušenia. Je založený na preddefinovanej množine pravidiel, definujúcej mieru rozpoznávania falošne negatívnych a falošne pozitívnych hrozieb. Infraštruktúra nástroja Suricata využíva prístup založený na detekcii viacerých vlákien, umožňujúcej efektívnejšie využívanie viacjadrových systémov a vykonávanie súbežnej analýzy sieťového prenosu, čím dokáže dosiahnuť väčšiu škálovateľnosť. Detekčný modul nástroja Suricata obsahuje viac vlákien, čo mu umožňuje presne alokovať výpočtový výkon a oddeliť operácie detekcie signatúr medzi niekoľkými vláknami [26].



Obrázok 7 Infraštruktúra IDPS nástroja Suricata

Na obrázku 7 je znázornená infraštruktúra systému detekcie a prevencie narušenia systému nástroja Suricata. Proces analýzy sieťového prenosu je možné vykonať buď prvým zachytením paketov priamo z karty sieťového rozhrania, alebo pomocou vopred zaznamenaného prenosu. Potom sa pakety dekódujú a následne v mechanizme opätovného zostavenia toku zostavia pakety do frontov toku. Aby bolo možné načítať paket, každé vlákno vyvolá funkciu Queue Handler, ktorá sa zaoberá načítaním a vyradením paketov vo vlákne. Užívateľ si tiež môže nakonfigurovať počet vlákien „Detekcia“ v závislosti od počtu jadier CPU, kde každé vlákno zodpovedá jednému jadru. Po porovnaní signatúr paketov s preddefinovanými signatúrami narušenia, a po rozhodnutí, či by určité pakety mali byť vylúčené alebo akceptované, sa vytvoria výstupné protokoly.

V systéme prevencie narušenia musí byť nástroj Suricata schopná vylúčiť pakety, ktoré obsahujú niečo, čo prenos označuje ako škodlivé. Najpoužívanejšou metódou snímania IPS je metóda Netfilter queue (NFqueue). Metóda NFqueue umožňuje nástroju Suricata vykonávať akcie ako *DROP* alebo *ACCEPT* na pakety, ktoré sa nachádzajú v sieťovom prenose.
Metóda NFqueue pracuje s nástrojom Suricata v režime IPS:

- prichádzajúci paket, ktorý zodpovedá pravidlu sa odosiela do nástroja Suricata prostredníctvom *nfnetlik,*
- nástroj Suricata dostane paket a rozhodne o ňom v závislosti od našej sady pravidiel,
- paket je prenášaný alebo odmietaný jadrom,
- na strane výkonu je počet NFqueue paketov obmedzený za sekundu v jednej fronte z dôvodu povahy komunikácie *nfnetlink* [27].

Pomocou metódy NFqueue môžeme v nástroji Suricata nastavovať nasledujúce možnosti:

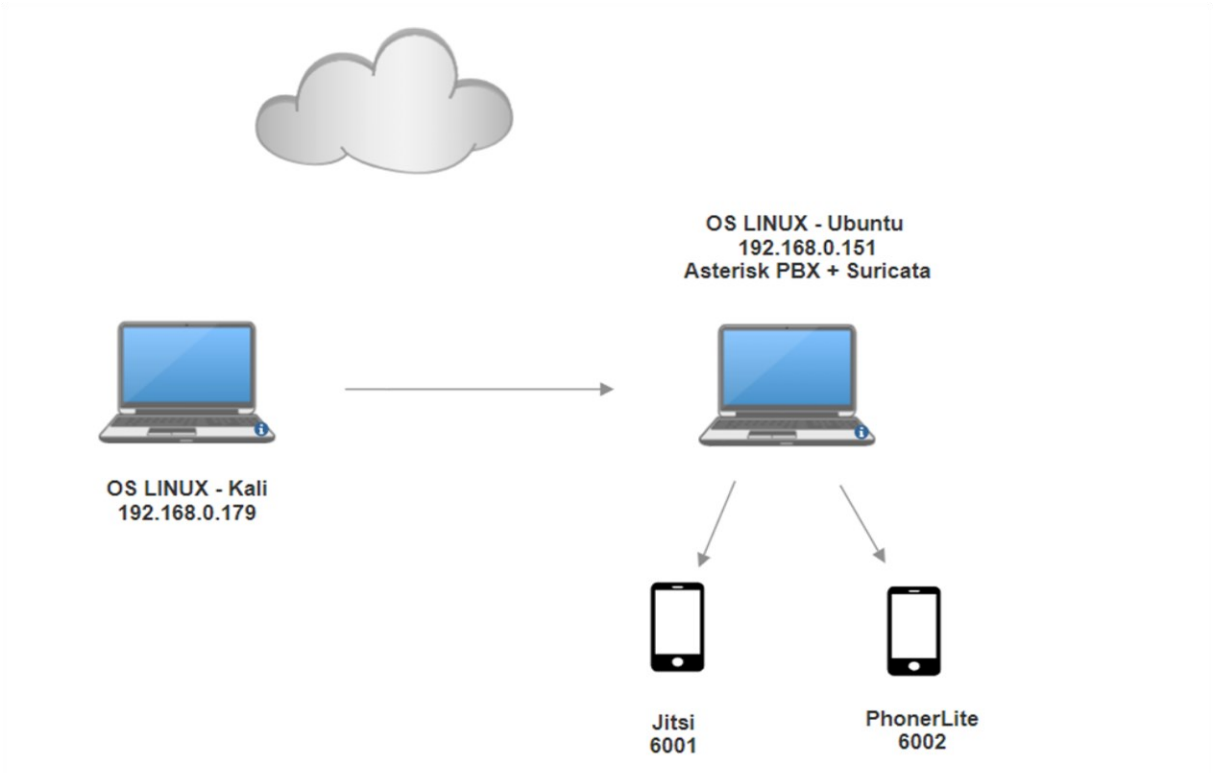
- - *queue-num:* číslo fronty,
- - *queue-balance*: paket je zaradený do fronty podľa rovnakých pravidiel,
- - *queue-bypass*: paket je prijatý, ak žiaden softvér nepočúva vo fronte,
- - *fail-open*: paket je prijatý v prípade, že je fronta plná,
- - *batching verdict*: verdikt sa pošle všetkým paketom [28].

5. Praktická časť

Praktická časť diplomovej práce je zameraná na predstavenie návrhu testovacej topológie, pre ktorú sme pomocou vhodných nástrojov pre simuláciu IP telefónnych hrozieb realizovali útoky na pobočkovú ústredňu Asterisk a testovali sme tak bezpečnostný nástroj Suricata, pracujúci v režime systému prevencie narušenia. Jednotlivé podkapitoly ďalej popisujú inštaláciu a konfiguráciu všetkých potrebných nástrojov pre realizovanie praktickej časti diplomovej práce.

5.1 Testovacia topológia

Pre testovanie IPS systémov nástrojom Suricata s filtračným systémom IPtables a NFtables, využívajúce metódu NFqueue som zvolila jednoduché zapojenie znázornené na obrázku 8. Topológia siete mohla byť realizovaná pomocou fyzického zapojenia, avšak berúc do úvahy epidemiologickú situáciu v krajine, súvisiacu so šíriacim sa infekčným ochorením COVID-19, sme po dohode s vedúcim diplomovej práce zvolili virtuálne zapojenie siete. Počítač s operačným systémom Linux, s verziou Kali 3.38.0 predstavuje útočníka, jeho IP adresa je 192.168.0.179. Druhý počítač s operačným systémom Linux, verziou Ubuntu 20.04.1 LTS obsahuje pobočkovú ústredňu PBX Asterisk verzie 18, na ktorú sú zapojené a nakonfigurované dva softvérové telefóny Jitsi (klapka 6001) a PhonerLite (klapka 6002) a tiež testovací nástroj Suricata. Pobočková ústredňa Asterisk a nástroj Suricata majú nastavenú rovnakú IP adresu, a to 192.168.0.151.



Obrázok 8 Topológia siete pre testovanie

5.2 Inštalácia Asterisk

Asterisk je voľne dostupná softvérová implementácia pobočkovej ústredne PBX, pracujúca na platformách operačného systému Linux a Unix. Vytvára rozhranie telefónnemu hardvéru, softvéru a tiež ľubovoľnej telefónnej aplikácie. Inštalačný balíček je voľne dostupný na oficiálnej webovej stránke [44]. Nižšie sú uvedené kroky, pomocou ktorých nainštalujeme a nakonfigurujeme Asterisk:

Pomocou nasledujúcich príkazov stiahneme a rozbalíme požadovanú verziu Asterisku.

 # wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-18-current.tar.gz
 # tar zxvf asterisk-18-current.tar.gz.

Po presunutí sa do rozbaleného adresára pomocou príkazu *cd asterisk-18.1.1* vykonáme inštaláciu prerekvizít a následnú konfiguráciu kódu pre kompiláciu a inštaláciu:

 # sudo contrib/scripts/install_prereq install
 # sudo ./configure
 # sudo make
 # sudo make install.

Nakoniec nainštalujeme vzorové súbory a konfigurácie:

 # sudo make samples
 # sudo make config
 # sudo ldconfig.

Pomocou nasledujúcich príkazov vytvoríme samostatného užívateľa a skupiny pre spustenie služieb a pridanie povolení:

 # sudo groupadd asterisk
 # sudo useradd -r -d /var/lib/asterisk -g asterisk asterisk
 # sudo usermod -aG audio,dialout asterisk
 # sudo chown -R asterisk.asterisk /etc/asterisk
 # sudo chown -R asterisk.asterisk /var/{lib,log,spool}/asterisk
 # sudo chown -R asterisk.asterisk /usr/lib/asterisk.

V ďalšom kroku nastavíme predvoleného užívateľa *asterisk:*

 # sudo nano /etc/default/asterisk
AST_USER="asterisk"
AST_GROUP="asterisk"

 # sudo nano /etc/asterisk/asterisk.conf
runuser = asterisk ;
rungroup = asterisk ;.

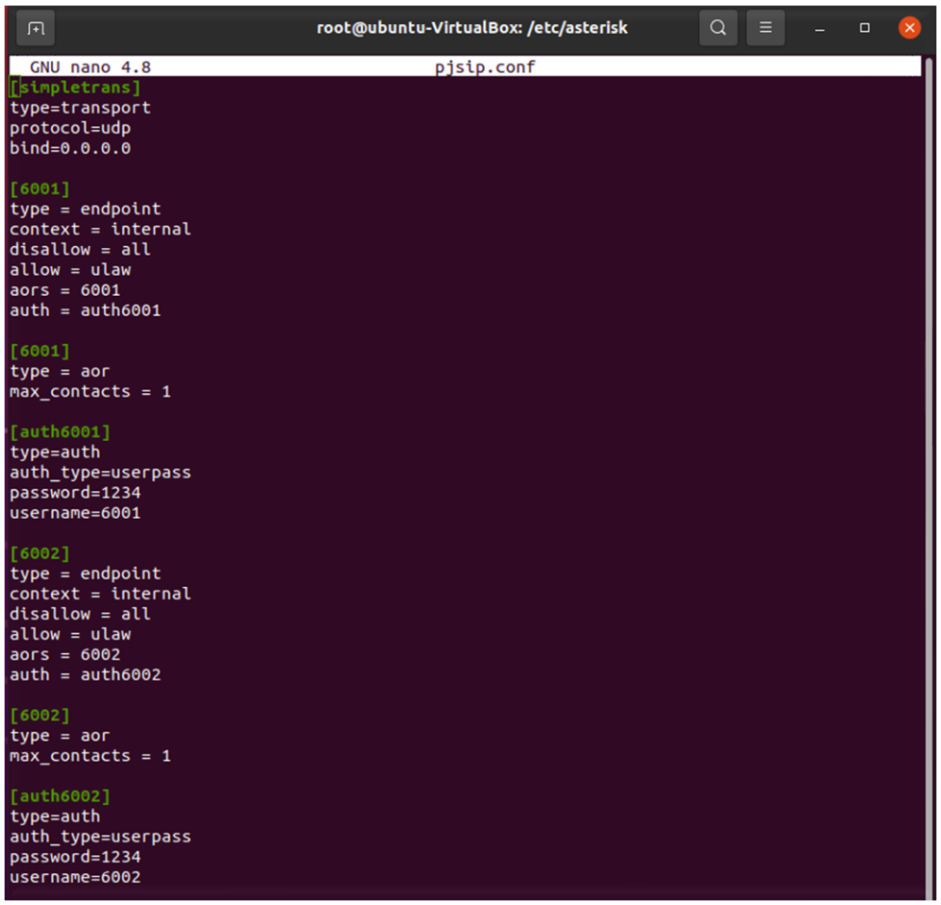
Pre spustenie Asterisku použijeme príkaz:

asterisk

5.2.1 Konfigurácia Asterisk

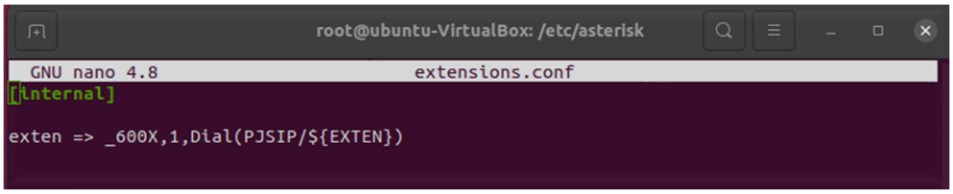
Konfigurácia ústredne Asterisk prebieha v adresári */etc/asterisk*, kde stačí nastaviť dva základné konfiguračné súbory, prvý konfiguračný súbor *pjsip.conf*, obsahujúci konfiguráciu zariadenia komunikujúceho s ústredňou Asterisk a druhý konfiguračný súbor *extensions.conf*, definujúci číslovací plán a teda správanie všetkých spojení v ústredni.

Na obrázku 9 je znázornený príklad minimálnej konfigurácie *pjpsip.conf* nadefinovaný pre dva telefóny.



Obrázok 9 Konfiguračný súbor pjsip.conf

Na obrázku 10 je znázornený príklad konfigurácie *extensions.conf*, definujúci číslovací plán, ktorý obsahuje informáciu pre smerovanie prichádzajúcich a odchádzajúcich hovorov. V tomto prípade pre volanie z a na klapky 6000 až 6009.

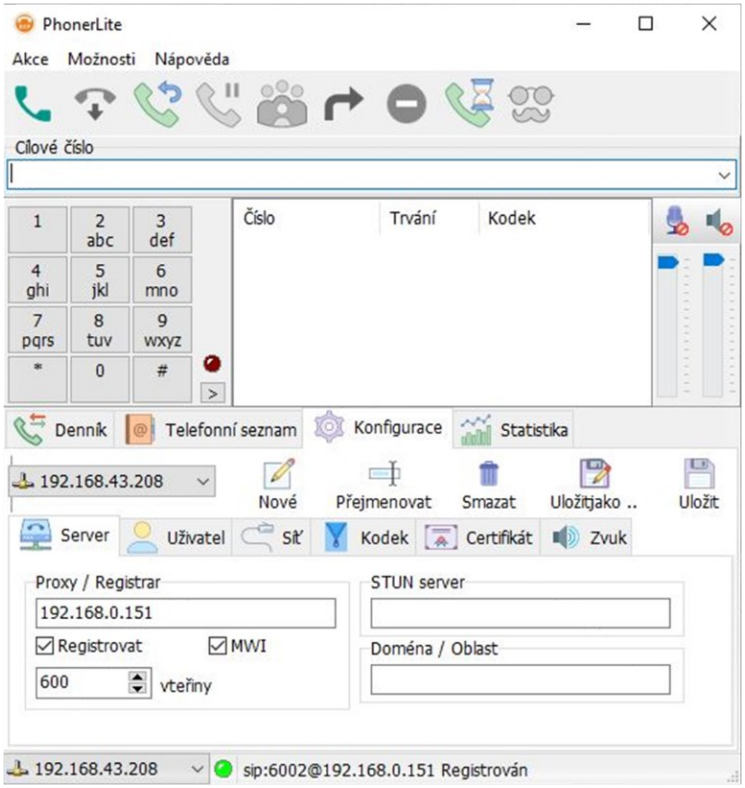


Obrázok 10 Konfiguračný súbor extensions.conf

5.2.2 Nastavenie telefónov

Pre overenie správnosti a funkčnosti ústredne Asterisk boli použité dva softvérové telefóny. Oba telefóny, prvý softvérový telefón *Jitsi* a druhý softvérový telefón *PhonerLite,* sú voľne dostupné k stiahnutiu z ich oficiálnych webových stránkach.

Po stiahnutí a inštalácií oboch telefónov je potrebné v *Nastaveniach* telefónu definovať *IP adresu serveru*, pre ktorý sa oba telefóny registrujú a *užívateľské meno* a *heslo*. Na obrázku 11 je znázornený príklad nastavenia softvérového telefónu *PhonerLite*. Rovnako sme postupovali aj v prípade nastavenia druhého softvérového telefónu *Jitsi*.



Obrázok 11 Nastavenie softvérového telefónu PhonerLite

5.3 Inštalácia nástroja Suricata

Nástroj Suricata je voľne dostupný k stiahnutiu na oficiálnej webovej stránke tohto nástroja. Pred inštaláciou je potrebné nainštalovať potrebné balíky nasledujúcimi príkazmi [50]:

 # apt-get install libpcre3 libpcre3-dbg libpcre3-dev build-essential libpcap-dev
 # apt-get install libyaml-0-2 libyaml-dev pkg-config zlib1g zlib1g-dev
 # apt-get install make libmagic-dev libjansson4 libjansson-dev
 # apt-get install python-yaml rustc cargo

 # apt-get install libnetfilter-queue-dev libnetfilter-queue1
 # apt-get install libnetfilter-log-dev libnetfilter-log1
 # apt-get install libnfnetlink-dev libnfnetlink0
 # apt-get install rustc cargo
 # cargo install --force --debug --version 0.14.1 cbindgen.

Následne vytvoríme adresár pre nástroj Suricata a jeho aktuálnu verziu nainštalujeme pomocou nasledujúcich príkazov:

 # mkdir ~/suricata_src
 # cd ~/suricata_src
 # sudo apt-get install suricata
 # wget https://www.openinfosecfoundation.org/download/suricata-6.0.1.tar.gz
 # tar xzvf suricata-6.0.1.tar.gz
 # cd suricata-6.0..

Po stiahnutí a rozbalení inštalačného balíčka súbor skompilujeme a nainštalujeme aj s konfiguračnými súbormi a pravidlami. V prípade, že má nástroj Suricata pracovať v režime IPS, povolíme ho nasledujúcim príkazom:

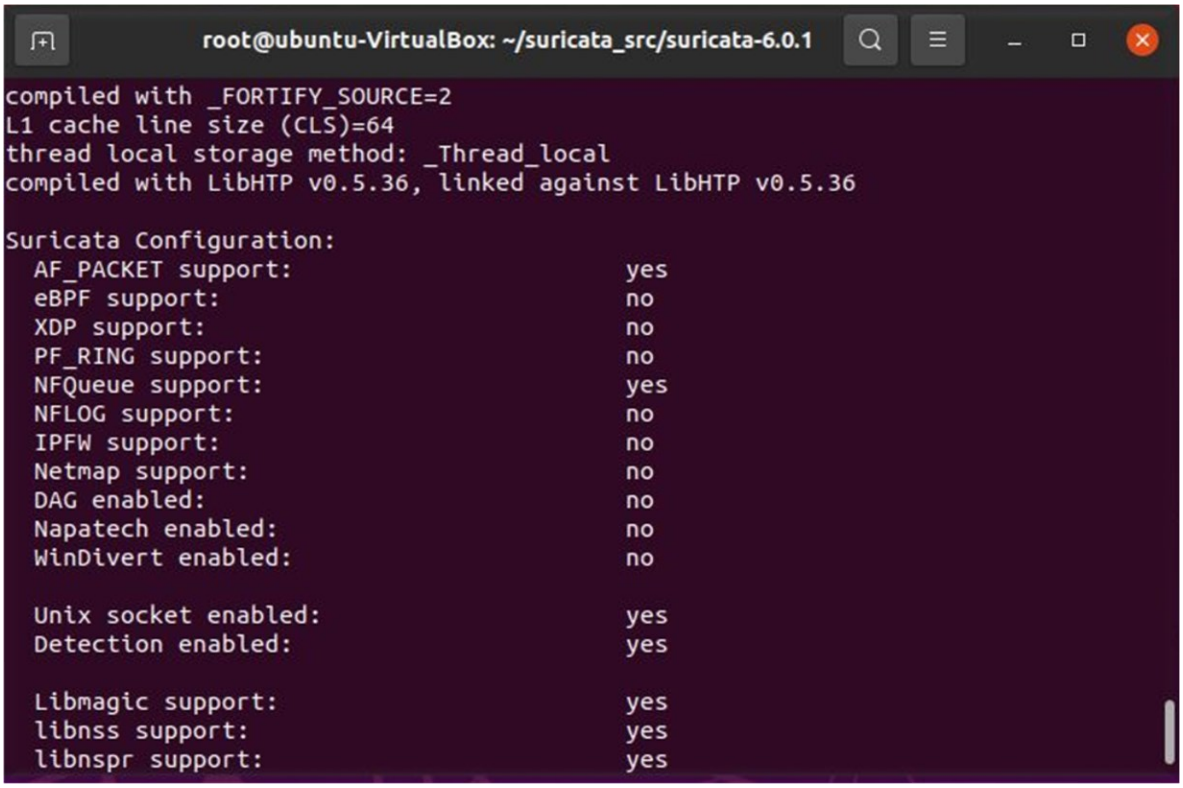
 # ./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --localstatedir=/var
 # ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var
 # make
 # make install
 # ./configure && make && make install-conf
 # ./configure && make && make install-rules.

Ak sa nástroj Suricata nenachádza v repozitári, môžeme ho pridať pomocou nasledujúcich príkazov:

 # sudo add-apt-repository ppa:oisf/suricata-stable
 # sudo apt update
 # sudo apt install suricata jq

V prípade, že chceme skontrolovať, či má nástroj Suricata povolený NFQ, urobíme tak pomocou príkazu:

 # sudo suricata --build-info.



Obrázok 12 Kontrola povolenia NFQ pre nástroj Suricata

Pre správne fungovanie nástroja Suricata je potrebné zmeniť IP adresu *HOME_NET: "[192.168.0.151]"* na IP adresu stroja, na ktorom sa nástroj Suricata nachádza. Zmenu je možné vykonať v konfiguračnom súbore */etc/suricata/suricata.yaml* pomocou príkazu:

 # sudo nano /etc/suricata/suricata.yaml.

Nastavenie filtračného nástroja IPtables, využívajúcej metódu NFqueues vykonáme pomocou nasledujúcich príkazov:

 # sudo iptables -I INPUT -j NFQUEUE --queue-bypass
 # sudo iptables -I OUTPUT -j NFQUEUE --queue-bypass

Pre uloženie jednotlivých pravidiel IPtables použijeme príkaz *sudo netfilter-persistent save*.

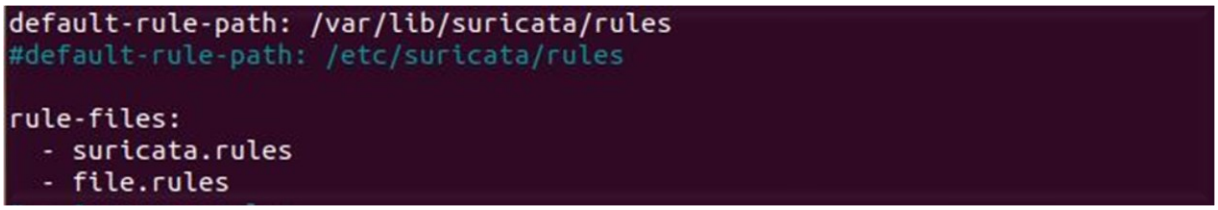
V prípade, že chceme zistiť, či sme pravidlá IPtables nastavili pre nástroj Suricata správne, môžeme vykonať kontrolu pomocou príkazu:

 # sudo iptables -vnL.

Obrázok 13 Kontrola nastavenia IPtables pre nástroj Suricata

V našom prípade, bolo potrebné pre správne fungovanie nástroja Suricata v IPS režime zmeniť v konfiguračnom súbore */etc/default/suricata* riadok *LISTENMODE=af-packet* na *LISTENMODE=nfqueue*.

Pravidlá pre testovanie nástroja Suricata boli automaticky vytvorené pri jeho inštalácii. Cesta k týmto pravidlám je definovaná v konfiguračnom súbore */etc/suricata/suricata.yaml* v časti *rule-files: - file.rules*.



Obrázok 14 Konfiguračný súbor file.rules

Nastavenie nástroja Suricata s filtračným systémom NFtables, využívajúcim NFqueues vykonáme pomocou nasledujúcich príkazov [52]:

 # sudo apt-get install nftables /inštalácia NFtables
 # nft -f /etc/nftables.conf /príkaz pre založenie tabuľky
 # nft list table inet filter /príkaz pre zobrazenie tabuľky
 # nft -i /prepnutie do NFtables
 # add chain inet filter firewall { type filter hook forward priority 0;} /vytvorenie chainu firewall
 # add chain inet filter IPS { type filter hook forward priority 10;} /vytvorenie chainu IPS
 # add rule inet filter IPS queue /naviazanie chainu IPS na NFqueue
 # add rule inet filter IPS queue num 3-5 fanout,bypass /naviazanie chainu IPS na NFqueue
 # nft list ruleset > /etc/nftables.conf /uloženie konfigurácie do súboru
 # systemctl enable nftables.service /povolenie služby NFtables

6. Realizácia praktickej časti

Realizácia praktickej časti je zameraná na vytváranie jednotlivých útokov, uskutočnených na pobočkovú ústredňu Asterisk, pre ktorú sú nakonfigurované dva softvérové telefóny PhonerLite a Jitsi. Nástroj Suricata je v prvej časti nastavený tak, aby pracoval s filtračným systémom IPtables a v druhej časti s jeho nástupcom NFtables. Pri realizácii útokov plní nástroj Suricata v oboch prípadoch funkciu prevencie narušenia systému a zaznamenáva jednotlivé pokusy pri podozrení na nebezpečnú komunikáciu v sieti. Pravidlá sú pri testovaní definované typom akcie *drop,* pri ktorom v prípade, že nástroj Suricata daný útok zaznamená, paket zahodí a vygeneruje upozornenie.

6.1 Testovanie nástroja Suricata v režime IPS - IPtables

6.1.1 SIPVicious

Nástroj SIPVicious predstavuje sadu nástrojov používaných pre skenovanie telefónnych systémov Voice over IP založených na protokole SIP. Obsahuje päť základných testovacích nástrojov, medzi ktoré patria nástroj *svmap* používaný ako SIP skener, nástroj *swvar* pre identifikáciu platných VoIP telefónov v sieti, nástroj *svcrack*, ktorý je schopný prelomiť heslá, ďalej nástroj *svreport* umožňujúci spravovať relácie vytvorené ostatnými nástrojmi, napríklad exportovať relácie do formátu pdf, xml a iné, a nástroj *svcrash* reagujúci na SIP správy svwar a svcrack so správou, ktorá spôsobuje zlyhanie starých verzií [46].

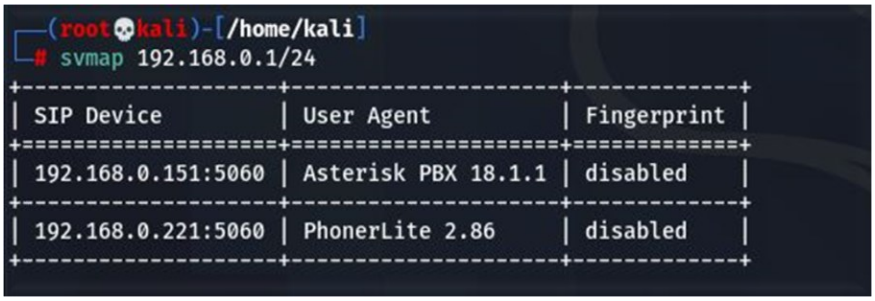
Pre správnu funkciu systému sú definované pre nástroj Suricata pravidlá, podľa ktorých zaznamenáva udalosti. Pravidlá sa nachádzajú v konfiguračnom súbore */etc/suricata/rules/files.rules*, vytvorenom automaticky pri inštalácii nástroja Suricata a tieto pravidlá môžeme podľa potreby upravovať. Pomocou pravidiel vykonávame porovnanie jednotlivých paketov, ktoré prechádzajú cez sieť a vyhodnocujú príslušné akcie. Pre testovanie útoku typu SIPVicious použitím príkazu *svmap* bolo potrebné upraviť akciu pravidla z *alert* na *drop* pravidlo:

drop udp $EXTERNAL_NET any -> $HOME_NET 5060 (msg:"ET SCAN Sipvicious User-Agent Detected (friendly-scanner)"; content:"|0d 0a|User-Agent|3A| friendly-scanner"; content:"OPTIONS";threshold: type limit, track by_src, count 5, seconds 120; reference:url,code.google.com/p/sipvicious/; reference:url,blog.sipvicious.org/; reference:url,doc.emergingthreats.net/2011716; classtype:attempted-recon; sid:2011716; rev:3; metadata:created_at 2010_07_30, updated_at 2010_07_30;)

Pomocou pravidla definujeme typ akcie *drop*, pri ktorom nástroj Suricata zahodí paket a vygeneruje upozornenie. Za pravidlom akcie *drop* nasleduje typ protokolu *udp* informujúci nástroj Suricata, ktorého protokolu sa pravidlo týka. Ďalšou časťou definujeme rozsah siete, pre ktoré bude pravidlo aplikované, pomocou dvoch premenných *$EXTERNAL_NET (IP adresa zdrojovej časti) a $HOME_NET (naša IP adresa definovaná pre testovanie)*. Ich hodnoty sú nastavené v konfiguračnom súbore suricata.yaml. Parametrom *any* definuje akýkoľvek zdrojový port z definovaného rozsahu, šípka -> udáva smer dátového toku a číslo *5060* značí port cieľovej IP adresy, na ktorom má byť pravidlo aplikované. Poslednou časťou je *msg,* ktorá vypíše text správy nástoja Suricata. V ňom je definovaný

typ *content*, pre ktorý definujeme názov metódy, depth znamená, koľko bytov bude od začiatku hlavičky kontrolovaných, *reference* obsahuje stránku týkajúcu sa pravidla*, classtype:attempted-recon, sid* a *rev* obsahujú klasifikáciu pravidla*.*

Pri testovaní bol použitý útok typu *svmap 192.168.60.1/24,* pomocou ktorého sme preskenovali celú sieť. Výsledok testu pri skenovaní siete odhalil, že v uvedenom rozsahu IP adries sa nachádza Asterisk PBX verzie 18.1.1. a softvérový telefón PhonerLite 2.86. Druhý softvérový telefón Jitsi tento príkaz neodhalil.



Obrázok 15 SIPVicious svmap útok

Pre zachytenie prechádzajúcich paketov cez sieťové rozhranie sme použili nástroj *tcpdump*, ktorý zachytil výsledok: *15:41:36.314475 IP 192.168.0.179.sip > 192.168.0.0.sip: SIP: OPTIONS sip:100@192.168.0.0 SIP/2.0*. Môžeme vidieť, že útok typu SIPVicious odosiela SIP správy typu *OPTIONS*. Tento typ SIP správy s metódou *OPTIONS* je okrem toho aplikovaný na pravidlo v konfiguračnom súbore v časti *contents.*

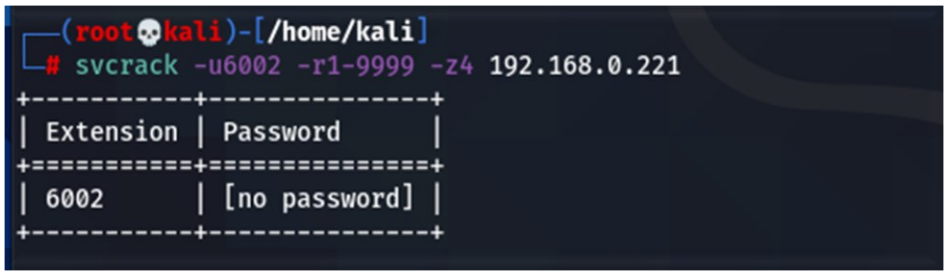
Nástroj Suricata zachytil útok typu SIPVicious a pri danej akcii došlo k zablokovaniu prenosu zo zdrojovej IP adresy na cieľovú adresu.

Výstup fast.log:
*04/24/2021-15:41:39.787802 [**Drop**] [**] [1:2011716:3] ET SCAN Sipvicious User-Agent Detected (friendly-scanner) [**] [Classification: Attempted Information Leak] [Priority: 2] {UDP} 192.168.0.179:5060 -> 192.168.0.151:5060*

Výstup z eve.json:
{
"timestamp": "2021-04-24T15:52:05.286209+0200",
"flow_id": 659484217274742,
*"event_type": "**drop**",*
*"src_ip": "**192.168.0.179**",*
"src_port": 5060,
*"dest_ip": "**192.168.0.151**",*
"dest_port": 5060,
"proto": "UDP",
"drop": {

```
   "len": 434,
   "tos": 0,
   "ttl": 64,
   "ipid": 56270,
   "udplen": 414
  },
  "alert": {
   "action": "blocked",
   "gid": 1,
   "signature_id": 2011716,
   "rev": 3,
   "signature": "ET SCAN Sipvicious User-Agent Detected (friendly-scanner)",
   "category": "Attempted Information Leak",
   "severity": 2
  }
}
```


V ďalšom teste bol vykonaný pokus o prelomenie hesla pomocou príkazu *svcrack -u6002 -r1-9999 -z4 192.168.0.221* pre softvérový telefón PhonerLite, ktorý sme pomocou útoku typu *svmap* zachytili v predchádzajúcom pokuse. Pomocou parametra *-u* sme definovali meno užívateľa softvérového telefónu PhonerLite, ktorého užívateľské meno bolo nastavené na hodnotu 6002, *-r* špecifikuje rozsah telefónnych čísel od 1 do 9999 a parametrom *-z4* definujeme počet miest pre heslo. Výsledkom pokusu je neodhalenie hesla pre zvolený telefón.



Obrázok 16 SIPVicious scrack útok

Pre testovanie útoku typu SIPVicious použitím príkazu *svcrack* bolo potrebné upraviť pravidlo pre nástroj Suricata zmenou akcie z *alert* na *drop* v konfiguračnom súbore */etc/suricata/rules/files.rules*:

drop udp $EXTERNAL_NET any -> $HOME_NET 5060 (msg:"ET SCAN Sipvicious Scan"; content:"OPTIONS"; threshold: type limit, count 5, seconds 10, track by_src; reference:url,blog.sipvicious.org; reference:url,doc.emergingthreats.net/2008578; classtype:attempted-recon; sid:2008578; rev:4; metadata:created_at 2010_07_30, updated_at 2010_07_30;)

Pomocou nástroja *tcpdump* sme pri testovaní zachytili rovnaký výsledok: *15:58:25.296958 IP 192.168.0.221.53688 > ubuntu-VirtualBox.sip: SIP: OPTIONS sip:192.168.0.151 SIP/2.0* ako v prípade prvého testu. Pakety prechádzajúce cez sieťové rozhranie používajú SIP správu typu *OPTIONS*.

Nástroj Suricata zachytil útok typu SIPVicious a pri danej akcii došlo k zablokovaniu prenosu zo zdrojovej IP adresy na cieľovú IP adresu.

Výstup fast.log:
*04/24/2021-15:58:01.270067 [**Drop**] [**] [1:2008578:4] ET SCAN Sipvicious Scan [**] [Classification: Attempted Information Leak] [Priority: 2] {UDP} 192.168.0.221:53688 -> 192.168.0.151:5060*

Výstup z eve.json:

```
{
 "timestamp": "2021-04-24T16:00:22.286160+0200",
 "flow_id": 40963845133494,
 "event_type": "drop",
 "src_ip": "192.168.0.221",
 "src_port": 53688,
 "dest_ip": "192.168.0.151",
 "dest_port": 5060,
 "proto": "UDP",
 "drop": {
  "len": 667,
  "tos": 0,
  "ttl": 128,
  "ipid": 9771,
  "udplen": 647
 },
 "alert": {
  "action": "blocked",
  "gid": 1,
  "signature_id": 2008578,
  "rev": 4,
  "signature": "ET SCAN Sipvicious Scan",
  "category": "Attempted Information Leak",
  "severity": 2
 }
```

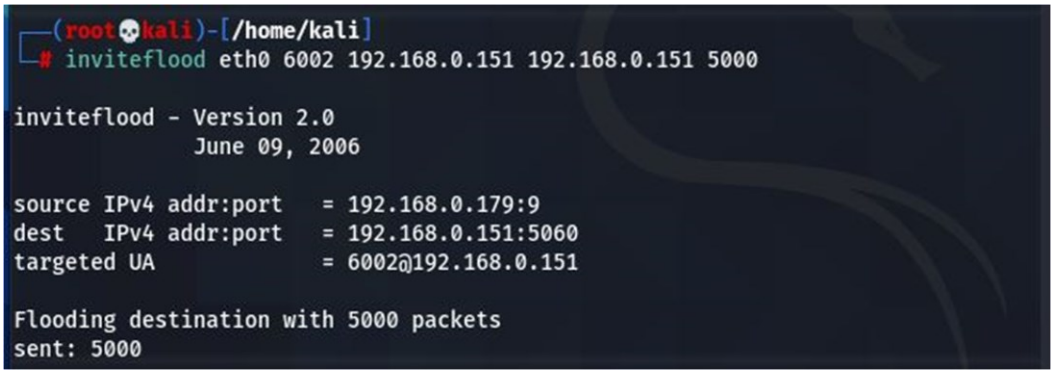

6.1.2 Inviteflood

Nástroj Inviteflood je používaný na zahltenie systému správami SIP/SDP INVITE prostredníctvom protokolu UDP/IP pre vykonávanie DoS útokov. Pomocou tohto nástroja môžeme cieľ zahltiť požiadavkou INVITE [47].

Pre testovanie útoku typu Inviteflood bolo potrebné upraviť pravidlo akcie z *alert* na akcie *drop* v konfiguračnom súbore */etc/suricata/rules/files.rules*:

drop udp any any -> any 5060 (msg:"GPL VOIP SIP INVITE message flooding"; content:"INVITE"; depth:6; threshold: type both, track by_src, count 100, seconds 60; classtype:attempted-dos; sid:2100158; rev:4; metadata:created_at 2010_09_23, updated_at 2010_09_23;)

Pri testovaní bol použitý útok typu *inviteflood eth0 6002 192.168.0.151 192.168.0.151 5000,* pomocou ktorého cez *eth0* rozhranie so SIP účtom *6002* telefónnej ústredne Asterisk s definovanými IP adresami nástroja Suricata bolo odoslaných 5*000* správ. Na obrázku 17 je znázornený útok, pri ktorom je počet INVITE paketov nastavený na 5000 tak, aby došlo k zaplaveniu cieľa. Počas realizácie útoku bude ústredňa Asterisk v podstate nepoužiteľná, pretože pre vytvorenie spojenia bude potrebovať o niečo dlhší čas.



Obrázok 17 Inviteflood útok

Pomocou nástroja *tcpdump* sme pri testovaní zachytili výsledok: *15:07:35.977494 IP 192.168.0.179.discard > ubuntu-VirtualBox.sip: SIP: INVITE sip:6002@192.168.0.151 SIP/2., kde vidíme, že pakety ktoré prechádzajú cez sieťové rozhranie* používajú SIP správu typu *INVITE*. Zo štatistiky súboru *stats.log* vidíme, koľko paketov bolo nástrojom Suricata prijatých a zablokovaných pred spustením útoku:

ips.accepted | Total | 3467
ips.blocked | Total | 10

Počet prijatých a zablokovaných paketov po spustení útoku:

ips.accepted | Total | 4802
ips.blocked | Total | 99

Nástroj Suricata zachytil útok typu Inviteflood a pri danej akcii došlo k zablokovaniu prenosu zo zdrojovej IP adresy na cieľovú adresu.

Výstup z fast.log:
04/03/2021-10:58:57.327619 ***[Drop]*** *[**] [1:2100158:4] GPL VOIP SIP INVITE message flooding [**] [Classification: Attempted Denial of Service] [Priority: 2] {UDP} 192.168.0.179:9 -> 192.168.0.151:5060*

Výstup z eve.json:


```
{
 "timestamp": "2021-04-03T11:08:46.346628+0200",
 "flow_id": 351234651087080,
 "event_type": "drop",
 "src_ip": "192.168.0.179",
 "src_port": 9,
 "dest_ip": "192.168.0.151",
 "dest_port": 5060,
 "proto": "UDP",
 "drop": {
  "len": 1094,
  "tos": 0,
  "ttl": 64,
  "ipid": 13192,
  "udplen": 1074
 },
 "alert": {
  "action": "blocked",
  "gid": 1,
  "signature_id": 2100158,
  "rev": 4,
  "signature": "GPL VOIP SIP INVITE message flooding",
  "category": "Attempted Denial of Service",
  "severity": 2
 }
```


6.1.3 NMAP

Nástroj NMAP patrí k jedným zo základných nástrojov, ktoré je možné použiť na skenovanie sietí. Môžeme ho použiť na vyhľadávanie živých hostiteľov v sieti, na skenovanie portov, detekciu operačného systému a podobne [48].

Cieľom testovania bolo aby nástroj Suricata bežiaci na rovnakej IP adrese ako je IP adresa pobočkovej ústredne Asterisk, v prípade, že sa útočník pokúsi v našom testovaní o skenovanie portov, tento test na základe pravidla zachytí a danú komunikáciu zablokuje. Pokus testu, pri ktorom sa útočník snaží vykonať pomocou príkazu NMAP skenovanie UDP portov pre rozsah siete 192.168.0.1/24 nebol úspešný. Použitie nižšie uvedeného príkazu, zobrazeného na obrázku 18, nástroj Suricata nezaznamenal žiaden výstup v súboroch eve.json a fast.log. V prípade, že skenovanie siete prebiehalo

na TCP portoch, nástroj Suricata pokus o podozrivú komunikáciu zaznamenal, ale neaktivoval pravidlo *drop* aby jej zabránil.

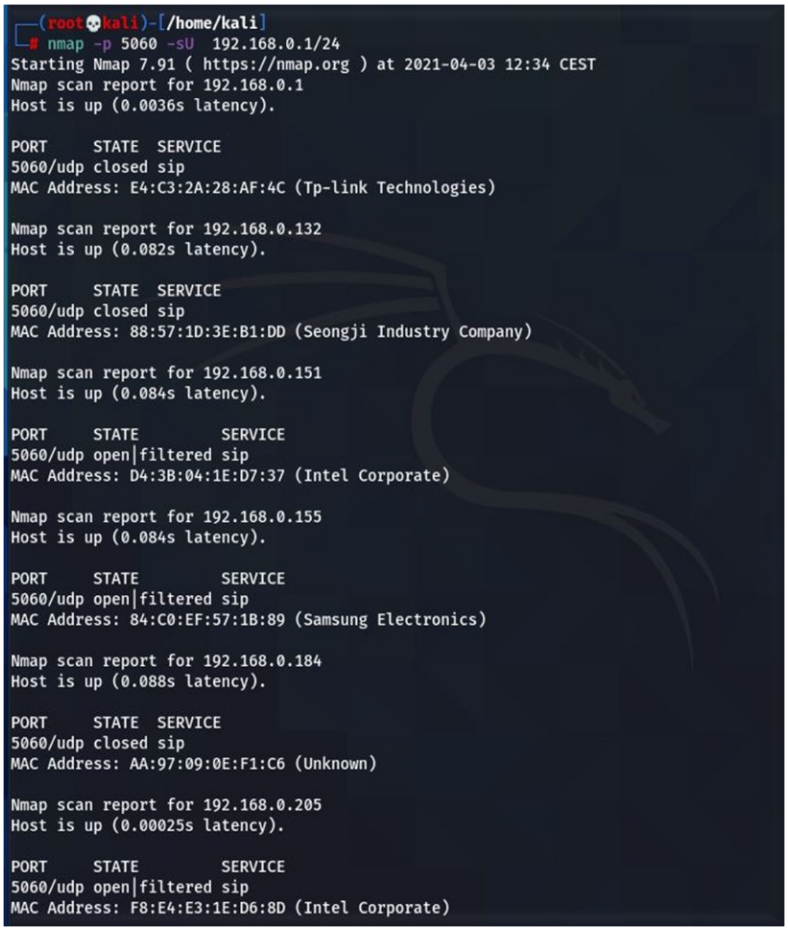
Pri testovaní boli použité a upravované pravidlá, ktorým som pri testovaní menila napríklad premenné *$EXTERNAL_NET* na IP adresu 192.168.0.179 a *$HOME_NET* na IP adresu 192.168.0.151. V rámci testovania som tiež vyskúšala zmeniť možnosti pravidiel, ako napríklad zmenu signatúr ID *sid*, verziu podpisu *rev,* vymazanie časti metadata a updated z konfiguračného súboru pravidiel */etc/suricata/rules/files.rules.* Ako príklad uvádzam nižšie uvedené pravidlá, na ktorých bola realizovaná zmena:

drop udp $EXTERNAL_NET any -> $HOME_NET any (msg:"ET SCAN NMAP -sS window 2048"; fragbits:!D; dsize:0; flags:S,12; ack:0; window:2048; threshold: type both, track by_dst, count 1, seconds 60; reference:url,doc.emergingthreats.net/2000537; classtype:attempted-recon; sid:2000537; rev:8; metadata:created_at 2010_07_30, updated_at 2010_07_30;)

drop udp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL SCAN PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:2100469; rev:4; metadata:created_at 2010_09_23, updated_at 2010_09_23;)

drop udp $EXTERNAL_NET any -> $HOME_NET 5060 (msg:"ET SCAN Smap VOIP Device Scan"; content:"<sip|3a|smap@"; offset:80; depth:40; reference:url,www.go2linux.org/smap-find-voip-enabled-devices; reference:url,doc.emergingthreats.net/2008526; classtype:attempted-recon; sid:2008526; rev:5; metadata:created_at 2010_07_30, updated_at 2010_07_30;)

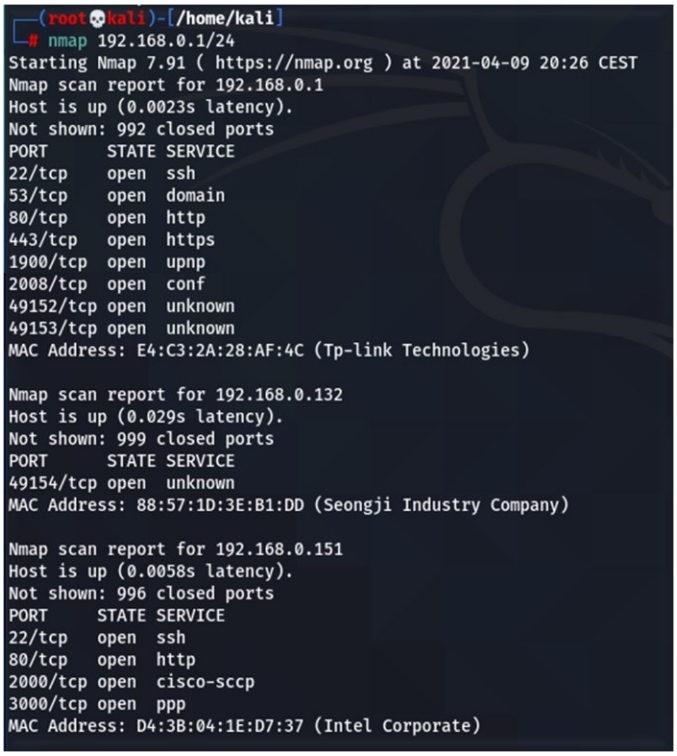
Pri testovaní bol použitý príkaz typu *nmap -p 5060 -sU 192.168.0.1/24,* kde príkazom *-p* definujeme *port 5060 a* príkazom *-sU* definujeme skenovanie UDP portu pre zadaný rozsah IP adries. Výsledok testu je znázornený na obrázku 18.



Obrázok 18 Skenovanie portov UDP pomocou NMAP

Pri testovaní boli použité typy pravidiel, ktoré boli automaticky stiahnuté pri inštalácii do konfiguračného súboru nástroja Suricata, ktoré podľa môjho názoru mali splniť to, že v prípade, že nástroj Suricata detekuje podozrivú komunikáciu, prenos zablokuje. Avšak pri testovaní pravidlá nesplnili ich funkciu a daný čas, kedy z počítača, ktorý predstavoval útočníka, nezaznamenal nástroj Suricata žiaden výstup pre *fast.log* a *eve.json*.

Pri testovaní príkazu NMAP som uskutočnila aj skenovanie TCP portov. Pri testovaní bol použitý príkaz nmap 192.168.0.1/24 pre preskenovanie celého rozsahu IP adresy. Výsledok testovania je znázornený na obrázku 19.



Obrázok 19 Skenovanie portov TCP pomocou NMAP

Pri testovaní skenovania TCP portov boli nastavené nasledujúce pravidlá v konfiguračnom súbore */etc/suricata/rules/files.rules* zmenou akcie *alert* na akciu *drop*:

Pravidlo:

drop tcp $EXTERNAL_NET any -> $HOME_NET 5800:5820 (msg:"ET SCAN Potential VNC Scan 5800-5820"; flow:to_server; flags:S,12; threshold: type both, track by_src, count 5, seconds 60;

reference:url,doc.emergingthreats.net/2002910; classtype:attempted-recon; sid:2002910; rev:6; metadata:created_at 2010_07_30, updated_at 2010_07_30;)

Výstup z fast.log:
*04/24/2021-16:16:18.909653 [**Drop**] [**] [1:2002910:6] ET SCAN Potential VNC Scan 5800-5820 [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.0.179:64563 -> 192.168.0.151:5810*

Pravidlo:
drop tcp $EXTERNAL_NET any -> $HOME_NET 5900:5920 (msg:"ET SCAN Potential VNC Scan 5900-5920"; flow:to_server; flags:S,12; threshold: type both, track by_src, count 5, seconds 60; reference:url,doc.emergingthreats.net/2002911; classtype:attempted-recon; sid:2002911; rev:6; metadata:created_at 2010_07_30, updated_at 2010_07_30;)

Výstup z fast.log:
*04/24/2021-16:16:18.807401 [**Drop**] [**] [1:2002911:6] ET SCAN Potential VNC Scan 5900-5920 [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.0.179:64563 -> 192.168.0.151:5910*

Pravidlo:
drop tcp $EXTERNAL_NET any -> $HOME_NET 5432 (msg:"ET SCAN Suspicious inbound to PostgreSQL port 5432"; flow:to_server; flags:S; s:S; threshold: type limit, count 5, seconds 60, track by_src; reference:url,doc.emergingthreats.net/2010939; classtype:bad-unknown; sid:2010939; rev:3; metadata:created_at 2010_07_30, former_category HUNTING, updated_at 2018_03_27;)

Výstup z fast.log:
*04/24/2021-16:16:18.895052 [**Drop**] [**] [1:2010939:3] ET SCAN Suspicious inbound to PostgreSQL port 5432 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.0.179:64563 -> 192.168.0.151:5432*

Pravidlo:
drop tcp $EXTERNAL_NET any -> $HOME_NET 1521 (msg:"ET SCAN Suspicious inbound to Oracle SQL port 1521"; flow:to_server; flags:S; s:S; threshold: type limit, count 5, seconds 60, track by_src; reference:url,doc.emergingthreats.net/2010936; classtype:bad-unknown; sid:2010936; rev:3; metadata:created_at 2010_07_30, former_category HUNTING, updated_at 2018_03_27;)

Výstup z fast.log:
*04/24/2021-16:16:18.899944 [**Drop**] [**] [1:2010936:3] ET SCAN Suspicious inbound to Oracle SQL port 1521 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.0.179:64563 -> 192.168.0.151:1521*

Pravidlo:
drop tcp $EXTERNAL_NET any -> $HOME_NET 1433 (msg:"ET SCAN Suspicious inbound to MSSQL port 1433"; flow:to_server; flags:S; threshold: type limit, count 5, seconds 60, track by_src;

reference:url,doc.emergingthreats.net/2010935; classtype:bad-unknown; sid:2010935; rev:3; metadata:created_at 2010_07_30, former_category HUNTING, updated_at 2018_03_27;)

Výstup z fast.log:
04/24/2021-16:16:18.864726 ***[Drop]*** *[**] [1:2010935:3] ET SCAN Suspicious inbound to MSSQL port 1433 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.0.179:64563 -> 192.168.0.151:1433*

Počas realizácie testovania sa okrem vyššie uvedených výsledkov objavil problém v tom, že skenovanie IP adries príkazom NMAP prebiehalo na komunikácii lokálneho *rozhrania [Priority: 3] {TCP} 127.0.0.1:9200 ->127.0.0.1:59098,* pričom nástroj Suricata nebol nastavený pre toto rozhranie. Pre odstránenie problému som zvolila použitie príkazu *iptables,* pomocou ktorého som povolila komunikáciu pre rozhranie *lo,* avšak výsledky testovania sa nezmenili.

6.2 Testovanie nástroja Suricata v režime IPS - NFtables

Druhá časť testovania nástroja Suricata v režime IPS bola zameraná na otestovanie novšieho nástupcu filtračného systému IPtables, a to NFtables. Konfigurácia tabuľky NFtables bola nastavená podľa postupu uvedeného na oficiálnej webovej stánke nástroja Suricata. Samotné nastavenie nástroja Suricata je podpísané v podkapitole 5.3. Výstup nastavenia je uvedený nižšie:

```
table inet filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }
        chain forward {
                type filter hook forward priority filter; policy accept;
        }
        chain output {
                type filter hook output priority filter; policy accept;
        }
        chain firewall {
                type filter hook forward priority filter; policy accept;
        }
        chain IPS {
                type filter hook forward priority filter + 10; policy accept;
                queue num 0
                queue num 3-5 bypass,fanout
        }
}
```

Pri realizácii pokusov o útok nástroj Suricata pomocou vyššie uvedeného nastavenia nedokázal zaznamenať žiadnu podozrivú komunikáciu v sieti. Pretože sa nástroj Suricata a pobočková ústredňa Asterisk nachádza na jednom stroji, bola pozmenená časť *chainu* pre *firewall* a *IPS* tak, že bolo

prepísané pravidlo „*hook forward*“ na „*hook input*“ pre filtrovanie prichádzajúcej komunikácie. Nižšie je zobrazený výstup tabuľky NFtables po zmene.

```
table inet filter {
        chain input {
                type filter hook input priority filter; policy accept;
        }
        chain forward {
                type filter hook forward priority filter; policy accept;
        }
        chain output {
                type filter hook output priority filter; policy accept;
        }
        chain firewall {
                type filter hook input priority filter; policy accept;
        }
        chain IPS {
                type filter hook input priority filter + 10; policy accept;
                queue num 0
                queue num 3-5 bypass,fanout
        }
}
```

Po konfiguračnej zmene v tabuľke NFtables sa pri pokuse o podozrivú komunikáciu vykonanú z počítača, ktorý predstavoval útočníka, začala v konfiguračnom súbore *fast.log* zaznamenávať komunikácia, avšak nástroj Suricata neidentifikoval konkrétny typ pokusu napríklad o útok pomocou SIPVicious alebo pomocou skenovania siete príkazom NMAP tak, ako bolo zachytávané pri testovaní, keď nástroj Suricata filtroval prenos pomocou IPtables.

Pri ďalšom testovaní som sa snažila zmeniť v časti *chain input {}* nastavenie *policy accept* na *policy drop* s cieľom, že pri filtrovaní prichádzajúcej komunikácie pakety zahodí a vygeneruje upozornenie. Tento pokus nebol realizovateľný, pretože v prípade nastavenia *policy accept* na *policy drop* došlo k odpojeniu od Internetového pripojenia a nebolo možné ďalej pokračovať v riešení.

V poslednej fáze testovania som sa snažila vytvárať pravidlá, ktoré som zapisovala do časti *chain input {}* s cieľom, že nástroj Suricata podozrivú komunikáciu zachytí, a danú akciu zablokuje, avšak neúspešne. Pokúsila som sa pridať napríklad pravidlá *nft add rule inet filter input ip saddr 192.168.0.1/24 counter drop, nft add rule inet filter input ip daddr 192.168.0.1/24 counter drop, nft add rule inet filter input ip saddr 192.168.0.151 counter drop* alebo *nft add rule inet filter input ip daddr 192.168.0.151 counter drop.* Nižšie je zobrazený výpis tabuľky NFtables pri použití jedného z pravidiel.

```
table inet filter {
        chain input {
                type filter hook input priority filter; policy accept;
                ip saddr 192.168.0.0/24 counter packets 1 bytes 183 drop
```

```
        }
        chain forward {
                type filter hook forward priority filter; policy accept;
        }
        chain output {
                type filter hook output priority filter; policy accept;
        }
        chain firewall {
                type filter hook input priority filter; policy accept;
        }
        chain IPS {
                type filter hook input priority filter + 10; policy accept;
                queue num 0
                queue num 3-5 bypass,fanout
        }
}
```

7. Zhodnotenie a záver

V diplomovej práci som sa venovala popisu hlasovej siete Voice over IP, kde som predstavila dva základné protokoly, používané v spojení s hlasovými službami, a to protokol pre komunikáciu v reálnom čase RTP a signalizačný protokol SIP. Oboznámila som sa s potenciálnymi útokmi a hrozbami, ktoré môžu byť v týchto sieťach realizované, ako aj s ich najčastejšie vyskytujúcimi sa typmi. Získala som informácie o tom, ako pracujú systémy prevencie narušenia IPS, a aké druhy existujú pre tieto systémy. Tiež som okrajovo popísala najčastejšie používané nástroje, pracujúce v režime prevencie narušenia. Z týchto systémov som sa detailnejšie zamerala na štúdium nástroja Suricata, ako aj na štúdium dvoch systémov pre filtrovanie sieťového prenosu, a to na systém IPtables a jeho nástupcu NFtables.

V praktickej časti diplomovej práce som sa najskôr venovala vytvoreniu pracoviska pre realizáciu testovania. Testovacia topológia tvorí jeden počítač, predstavujúci útočníka, na ktorom je nainštalovaná verzia Kali operačného systému Linux. Druhý počítač s operačným systémom Linux verzie Ubuntu obsahuje nástoj Suricata a pobočkovú ústredňu Asterisk, na ktorú boli nainštalované dva softvérové telefóny. V praktickej časti popisujem detailnú konfiguráciu nástroja Suricata, pracujúceho v režime prevencie narušenia systému, najskôr v spojení s filtračným systémom IPtables a neskôr s jeho novšou verziou NFtables. Obsahuje tiež inštaláciu a konfiguráciu pobočkovej ústredne Asterisk a nastavenie dvoch softvérových telefónov Jitsi a PhonerLite.

Cieľom bolo otestovať a overiť ako pracuje nástroj Suricata v režime IPS, ktorý filtruje sieťový prenos pomocou filtračných systémov IPtables a NFtables a preposiela ho do fronty NFqueue, a v prípade, že zachytí podozrivú komunikáciu prenos zablokuje a vygeneruje upozornenie.

Pri testovaní nástroja Suricata v režime IPS s filtračným systémom IPtables boli použité útoky typu SIPVicious, Inviteflood a skenovanie siete pomocou príkazu NMAP. Pred spustením daného pokusu o útok bolo potrebné najskôr v konfiguračnom súbore nástroja Suricata upraviť pravidlá, ktoré boli automaticky stiahnuté pri inštalácii tohto nástroja. Pre úspešné zaznamenanie útoku typu SIPVicious nástrojom Suricata bolo potrebné zmeniť typ akcie z *alert* na *drop* rovnako ako pre útok pomocou Inviteflood. V prípade pokusu o testovanie príkazom NMAP bolo cieľom vykonať skenovanie UDP portov 5060/5061 pre nástroj Suricata, čo však nebolo úspešné. Zrealizovala som niekoľko zmien pravidiel, vrátane zmeny akcie a sidu, tiež som sa pokúsila o vytvorenie vlastného pravidla tým, že som pozmenila pravidlá, ktoré fungovali pri skenovaní TCP portov pomocou príkazu NMAP. Pre všetky tri prípady bolo testovanie vykonané niekoľkokrát a vždy s rovnakým výsledkom, a to pri dvoch pokusoch o útok nástroj Suricata zaznamenal podozrivú komunikáciu a daný paket zahodil a vygeneroval upozornenie. V prípade testovania príkazom NMAP nástroj nezaznamenal žiadnu podozrivú komunikáciu a nedošlo k zablokovaniu sieťového prenosu.

V ďalšej časti realizácie praktickej časti diplomovej práce mal byť nástroj Suricata testovaný s použitím filtračného systému NFtables. V tejto fáze testovania som predpokladala, že sa mi potvrdí domnienka, že nástroj Suricata v režime IPS bude v spojení s filtračným systémom NFtables jednoduchší na vytváranie pravidiel pre filtrovanie sieťového prenosu. Avšak, pri konfigurácii sa mi nepodarilo systém NFtables nastaviť tak, aby daný pokus o útok zachytil, zablokoval a vygeneroval upozornenie tak, ako v prípade, keď som použila nástroj Suricata s filtračným systémom IPtables.

Na základe zrealizovaného testovania sa domnievam, že testovanie nástroja Suricata v režime IPS a s využitím filtračného systému IPtables je náročnejšie na vytváranie pravidiel pre správne zachytávanie podozrivej komunikácie, než pri použití filtračného systému NFtables, kde môj predpoklad bol, že stačí vytvoriť jednoduché pravidlo do tabuľky, ktoré by túto podozrivú komunikáciu

v prípade potreby zablokovalo. V budúcnosti by sa dalo pokračovať v testovaní filtračného systému NFtables v spojení s nástrojom Suricata, ktorý pracuje v režime IPS.

Zdroje

[1] Ashon A. Syed, Ilyas Mohammad. *VoIP Handbook: Applications, Technologies, Reliability, and Security.* Publisher: CRC Press, 2008. ISBN: 9781420070200.

[2] Nguyen H Nam. *Essential Cyber Security Handbook.* Publication date 2016.

[3] Thermos Peter, Takanen Ari. *SECURING VoIP NETWORKS. THREATES,VULNERABILITIES AND COUNTERMEASURES.* Boston: Pearson Education, Inc., 2008. ISBN-13: 978-0-321-43734-1.

[4] Gough Michael. *How to Cheat at VoIP Security.* Rockland: Syngress Publishing, Inc., 2007. ISBN 10: 1-59749-169-1.

[5] Vacca R. John. *Computer and Information Security Handbook*. Burlington: Elsevier Inc., 2009. ISBN: 978-0-12-374354-1.

[6] Goar Vishal, Kuri Manoj, Kumar Rajesh, Senjyu Tomonobu. *Advances in Information Communication Technology and Computing.* Publisher: Springer Nature Singapore Pte Lrd., 2019. ISBN: 978-981-15-5420-9.

[7] COLLIER, Mark a David ENDLER. *HACKING EXPOSED™: Unified Communications & VoIP Security Secrets & Solutions*. Second Edition. United States: McGraw-Hill Education, 2014. ISBN 978-0-07-179877-8.

[8] COLLIER, Mark a David ENDLER. *HACKING EXPOSED VoIP: Voice Over IP Security Secrets & Solution.* Osborne: Mc-Graw-Hill, 2007. ISBN: 9780072263640.

[9] Check Point SOFTWARE TECHNOLOGIES LTD. *Intrusion Prevention System* [online]. Dostupné z: https://www.checkpoint.com/products/intrusion-prevention-system-ips/

[10] Barracuda. *What is an Intrusion Prevention System?* [online]. Dostupné z: https://www.barracuda.com/glossary/intrusion-prevention-system

[11] ITProToday. *NIPS and HIPS*. [online]. Dostupné z: https://www.itprotoday.com/security/nips-and-hips

[12] Technopedia. *Network-based Intrusion Prevention System (NIPS).* [online]. Dostupné z: https://www.techopedia.com/definition/4030/network-based-intrusion-prevention-system-nips

[13] PhoenixNAP. *Iptables Tutorial: Ultimate Guide To Linux Firewall*. [online]. Dostupné z: https://phoenixnap.com/kb/iptables-tutorial-linux-firewall

[14] Er Ravudra Pawadia. *IPTables Tutorial: Beginners to Advanced Guide To Linux Firewall*. [online]. Dostupné z: https://erravindrapawadia.medium.com/iptables-tutorial-beginners-to-advanced-guide-to-linux-firewall-839e10501759

[15] Ellingwood Justin. Digitalocean. *A Deep 22Dive into IPtables and Netfilter Architecture.* [online]. Dostupné z: https://www.digitalocean.com/community/tutorials/a-deep-dive-into-iptables-and-netfilter-architecture

[16] Calyptix. *Intrusion detection and prevention systems: IDS IPS overview*. [online]. Dostupné z: https://www.calyptix.com/intrusion-detection-and-prevention-systems-ids-ips-overview/

[17] Apriorit. *Modifying Network Traffic with NFQUEUE and ARP Spoofing*. [online]. Dostupné z: https://www.apriorit.com/dev-blog/598-linux-mitm-nfqueue

[18] Flylib.com. *IDS and IPS Architecture*. [online]. Dostupné z: https://flylib.com/books/en/2.352.1/ids_and_ips_architecture.html

[19] RANGEFORCE. *Suricata as an IPS.* [online]. Dostupné z: https://materials.rangeforce.com/tutorial/2020/02/09/Suricata-Intrusion-Prevention-System/

[20] BRICATA. *What is Suricata? Intro to a Best of Breed Open Source IDS and IPS.* [online]. Dostupné z:
https://bricata.com/blog/what-is-suricata-ids/
[21] OISF. *Suricata User Guide. Release 7.0.0-dev.* [online]. Dostupné z:
https://readthedocs.org/projects/suricata/downloads/pdf/latest/
[22] ResearchGate. *A Survey on Network Security Monitoring Implementations.* [online]. Dostupné z:
file:///C:/Users/42077/Downloads/ASurveyonNetworkSecurityMonitoringImplementations.pdf
[23] DATAPACKET. *Securing your server with nftables.* [online]. Dostupné z:
https://blog.datapacket.com/securing-your-server-with-nftables/
[24] IPtables vs. Nftables. *What is nftable, and how is it different from IPtables?* [online]. Dostupné z:
https://ungleich.ch/en-us/cms/blog/2018/08/18/iptables-vs-nftables/
[25] Linux Audit. *Diferences between iptables and nftables explained.* [online]. Dostupné z:
https://linux-audit.com/differences-between-iptables-and-nftables-explained/
[26] Fekolkin Roman. *Intrusion Detection and Prevention Systems: Overview of Snort and Suricata*. [online]. Dostupné z:
https://www.researchgate.net/publication/297171228_Intrusion_Detection_and_Prevention_Systems_Overview_of_Snort_and_Suricata
[27] Longo Giuseppe, Leblond Eric. *Suricata IDPS and its interaciton with Linux kernel*. [online]. Dostupné z:
https://netdevconf.info/1.1/proceedings/papers/Suricata-IDPS-and-its-interaction-with-Linux-kernel.pdf
[28] STAMVS NETWORK. *SURICATA Mixing IPS/IDS Mode.* [online]. Dostupné z:
https://openisf.files.wordpress.com/2015/11/suricata_mixed_mode_g-longo.pdf
[29] Federal Communications Commission. *Voice Over Internet Protocol (VoIP).* [online]. Dostupné z:
https://www.fcc.gov/general/voice-over-internet-protocol-voip
[30] 3CX. *What is Voice over IP.* [online]. Dostupné z:
https://www.3cx.com/pbx/voice-over-ip/
[31] JavaTpoint. *Voice Over Internet Porotocol (VoIP).* [online]. Dostupné z:
https://www.javatpoint.com/voice-over-internet-protocol
[32] RFC3550. *RTP: A Transport Protocol for Real-Time Applications.* [online]. Dostupné z:
https://tools.ietf.org/html/rfc3550
[33] Koistinen Tommi. Nokia Telecommunications. *Protocol overview: RTP and RTCP.* [online]. Dostupné z:
https://www.researchgate.net/publication/251203018_Protocol_overview_RTP_and_RTCP
[34] RUSSELL, Travis. SESSION INITIATION PROTOCOL (SIP), Controlling Convergent Networks. United States: The McGraw-Hill Companies, 2008. ISBN 0-07-164367-2.07-164367-2.
[35] RFC3261. *SIP: Session Initiation Protocol.* [online]. Dostupné z:
https://tools.ietf.org/html/rfc3261#section-1
[36] AT&T Business. *Open Source IDS Tools: Comparong Suricata, Snort, Bro (Zeek), Linux.* [online]. Dostupné z:
https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview
[37] Snort. *New to Snort?* [online]. Dostupné z: https://www.snort.org/
[38] Cisco. *Snort IPS*. [online]. Dostupné z:

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_utd/configuration/xe-16-12/sec-data-utd-xe-16-12-book/snort-ips.pdf
[39] TechRepublic. *Using Snort for intrusion detection.* [online]. Dostupné z: https://www.techrepublic.com/article/using-snort-for-intrusion-detection/
[40] Bricata. *Zeek IDS [formerly known as Bro] is One of the Most Powerful Cybersecurity Tools You've Never Heard Of.* [online]. Dostupné z: https://bricata.com/blog/zeek-ids-threat-detection/
[41] Zeek. *Introduction.* [online]. Dostupné z: https://docs.zeek.org/en/current/intro/
[42] Gupta Alka, *Sharma Sen Lalit. Performance Evaluation of Snort and Suricata Intrusion Detection Systems on Ubuntu Server.* [online]. Dostupné z: https://www.researchgate.net/publication/337446926_Performance_Evaluation_of_Snort_and_Suricata_Intrusion_Detection_Systems_on_Ubuntu_Server
[43] DNSstuff. *8 Best HIDS Tools—Host-Based Intrusion Detection Systems.* [online]. Dostupné z: https://www.dnsstuff.com/host-based-intrusion-detection-systems
[44] Asterisk. [online]. Dostupné z: https://www.asterisk.org/
[45] Logz.io. *Grafana vs. Kibana: The Key Differences to Know.* [online]. Dostupné z: https://logz.io/blog/grafana-vs-kibana/
[46] Pypi.org *Welcome to SIPVicious security tools.* [online]. Dostupné z: https://pypi.org/project/sipvicious/
[47] ThreatRavnes. *Inviteflood – Tool Used to Perform DOS Attack on VOIP Network*. [online]. Dostupné z: https://threatravens.com/inviteflood-tool-used-to-perform-dos-attack-on-voip-network/
[48] Varonis. *How to Use Nmap: Commands and Tutorial Guide.* [online]. Dostupné z: https://www.varonis.com/blog/nmap-commands/
[49] LinOxide. *How to Install ELK on Ubuntu 20.04.* [online]. Dostupné z: https://linoxide.com/ubuntu-how-to/install-elk-on-ubuntu/
[50] Suricata. *Installation.* [online]. Dostupné z: https://suricata.readthedocs.io/en/suricata-6.0.1/install.html
[51] INFOSEC WRITE-US. *Secure network monitoring with elastic – Packetbeat + Suricata.* [online]. Dostupné z: https://medium.com/bugbountywriteup/secure-network-monitoring-with-elastic-packetbeat-suricata-7ecd871b1a52
[52] Suricata. *Setting up IPS/inline for Linux.* [online]. Dostupné z: https://suricata.readthedocs.io/en/suricata-6.0.1/setting-up-ipsinline-for-linux.html
[53] VOCUS. *Best practice to reduce the risk of toll fraud*. [online]. Dostupné z: https://www.vocus.com.au/news/best-practice-to-reduce-the-risk-of-toll-fraud
[54] Vidyo. *Detect & Protect Against Wangiri Callback Fraud.* [online]. Dostupné z: https://www.enghousenetworks.com/enghouse-resources/blog/detect-protect-against-wangiri-callback-fraud/